Polynote vs. Jupyter

Prepared by :

- Amit Samra
- Preeti Sehgal
- James Thompson

Origin

- Computational Notebooks are virtual environments used for coding. Collection of ordered cells.
    - 1988 Wolfram Mathematica on Mac
    - Now MATLAB, Python, Julia, Scala, SQL and more.
- Open-source notebook for data science / machine learning.
- Created by Netflix to overcome shortcomings of existing notebook tools for Scala.

Special Features

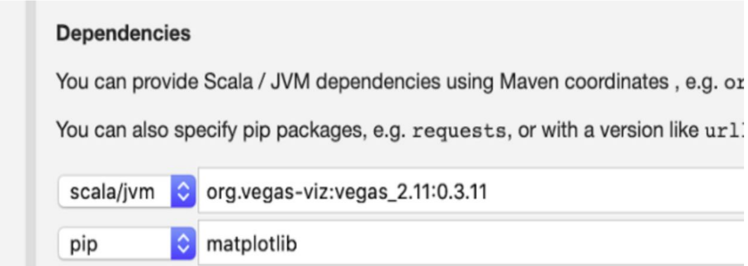
- Reproducibility – cells position is taken into account when executing (no hidden cells). Keeping track of variables defined in each cell > principle of least surprise.
- Visibility – expanded view of kernel by highlighting currently-running cell code without going into logs.
- For Polyglots but emphasis on Scala.
- Editing Improvements: IDE-like features such as autocomplete, parameter hints and error highlighting.
- Native data exploration and visualization.
- Dependency and configuration management – setup saved within notebook itself rather than external files.

Comparison to Jupyter

- Open source.
- Data Science / Machine Learning.
- Polyglot.
  - Jupyter: Python, Ruby, R
  - Polynote: Python, Scala, SQL
- Visualization via matplotlib.

Contrast to Jupyter

- In-Notebook Multiple Language Support*
  - Polynote can support Python, Scala, and SQL languages at the same time in one notebook.
  - It does this by sharing the different language cells variables between cells in the notebook, enabling different languages to work seamlessly together
- Reproducibility of Code
  - A big difference is that Juypter uses REPL where Polynote is more like writing a script
  - Results in consistent code, better shareability, but less flexibility
- Built in Visualization and Data Awareness
  - No need to pip Install
  - GUI to viewdata and value of variables
  - Progress window



Dependencies

You can provide Scala / JVM dependencies using Maven coordinates , e.g. or

You can also specify pip packages, e.g. `requests`, or with a version like `url`

scala/jvm: org.vegas-viz:vegas_2.11:0.3.11

pip: matplotlib

Use cases best suited for Polynote

(where you might use this instead of Jupyter)

- Heavy requirements for data visualization.
  - Polynote is best suited if the requirements are heavy on data visualization.
  - It has a Built-in data visualization GUI which means less code to write and faster turnarounds
- Requirement for multiple languages in a single notebook
  - Polynote allows multiple languages in one notebook
  - Easier to collaborate with resources using different languages for the same deliverable
- The sequence of cell execution is critical
  - Polynote is rigid on the sequence of cell execution
  - Reduces possible errors due to out of sequence execution of cells
- Heavy requirement for text outputs
  - Polynote has WYSIWIG (**W**hat **Y**ou **S**ee **I**s **W**hat You **G**et) editor
  - Reduces time for testing text outputs

Recommendation

It’s a relatively new tool and what makes or breaks a product is the network effect. It may not be worthwhile investing in a new technology that has yet to be proven outside Netflix.

Sources

https://en.wikipedia.org/wiki/Notebook_interface

https://polynote.org/

https://github.com/polynote/polynote

https://netflixtechblog.com/open-sourcing-polynote-an-ide-inspired-polyglot-notebook-7f929d3f447

https://medium.com/@brianray_7981/did-netflix-just-kill-databricks-with-releasing-polynote-for-free-c2d1f6dc0da3

https://hackernoon.com/what-you-need-to-know-about-netflixs-jupyter-killer-polynote-gb2a27yd

https://stackoverflow.com/questions/39008069/r-and-python-in-one-jupyter-notebook

https://towardsdatascience.com/what-you-need-to-know-about-netflixs-jupyter-killer-polynote-dbe7106145f5