Jupyter on Steroids

ABSTRACT

Despite the plethora of recently proposed tools and frameworks, common data analysis tasks such as data retrieval, curation and visualization remain laborious and non-trivial. The limited flexibility of existing frameworks, pushes code-literate data analysts to interactive notebooks such as Jupyter.

In this work, we argue that interactive notebooks are suboptimal for commonly performed tasks, with regards to their ease of use and interactivity. We propose ViDeTTe, a framework designed to facilitate data analysis through the utilization of a template language. We implement a sample data analysis work flow and present the benefits of using ViDeTTe instead of commonly used imperative language in iPython and Jupyter notebooks.

1. INTRODUCTION

User-friendly data analysis tools and frameworks often provide limited flexibility, as they usually focus on a predetermined set of use/analysis cases or a small fraction of the typically large data analysis pipeline. This lack of flexibility often pushes code-literate analysts towards the use of interactive notebooks such as Jupyter.

Interactive notebooks allow the use of popular, high-level and highly expressive imperative languages, such as Python, for analyzing data and composing the results into an easily readable notebook-like interface. Due to the wide popularity of such languages, there is also a huge collection of third-party libraries that can be used by data scientists as building blocks of a much bigger analytical process. Furthermore, the web environment of notebooks enables collaboration between data scientists, since it allows them to directly interact with the user interface in order to develop and run code, process data, generate visualizations, and lastly, compose their findings into an interactive (and re-runnable) report-like page, that contains code, visualizations and textual description of the analysis.

However as we show in this work, interactive notebooks are still suboptimal with regard to ease of use and interactivity. Setting up notebook environments and dependencies, obtaining and combining data and generating the respective visualizations, requires technical knowledge that often exceeds the skill-set of a typical data scientist. Lastly, while such notebooks support the generation of interactive visualizations, this interactivity is not an integral part of the data analysis process.

We address these issues, by extending interactive notebooks with a template language called ViDeTTe. The main contributions of this extension are:

- *Declarative semantics:* ViDeTTe implements formal declarative *Model-View-View-Model* (MVVM) semantics.

  Fill in why this is a good thing. I have no idea.

- *Expressive template language:* Prior database work, treats a page as a database view. Building on that, our template language goes beyond SQL query and view definition in both style and fundamental expressiveness. It is a mixture of query as well as web templating language that works on ordered (arrays) and semi-ordered (JSON) data.

- We allow in-line declarative code directly in JSON...

In this work, we demonstrate the use of ViDeTTe via a walkthrough example. Specifically, we want to use website access data, plot an access count histogram, as well as the recorder user demographics (age groups). We then want to interact with the histogram plot and select a time region. We want this action to automatically update the second plot with the user demographics in the selected time window. We assume a Jupyter server, where the analysts develop their notebooks and a different database server where data is stored. To retrieve the entirety of the required data, we have to query two different databases and join the returned JSON files. Figure ?? shows how our databases are organized. Our fictional analyst will perform the following tasks:

- Data retrieval from remote databases.
- Data curation: Join data and prepare for visualization.
- Data visualization.

The remainder of this paper is organized as follows: Sections 3 – 5 present a direct comparison of using ViDeTTe

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

and an imperative language such as Python in order to complete the tasks of our example. Throughout these sections, we demonstrate some of the main contributions of ViDeTTe. Section 6 provides further discussion regarding our proposed extension and presents other useful aspects of it not used in our walkthrough. Finally, Section 7 concludes the paper.

2. DATA RETRIEVAL

> *It might be a good idea to merge these 3 sections into 1 called "Walkthrough example" or something like that. Depends on how large these sections become.*

3. DATA CURATION

DATA CURATION

4. VISUALIZATION

VISUALIZATION

5. DISCUSSION

DISCUSSION

6. CONCLUSION

7. REFERENCES