Sloan Digital Sky Survey Software Development: Week 9 Progress Report

Riley Thai, Andy Casey (Supervisor)

January 25, 2024

1 Aims and Objectives

1. Develop a out-of-memory web application for multi-dimensional exploration of SDSS-V data.
2. Work on and contribute to the development of open-source Python libraries.

2 Background

The Sloan Digital Sky Survey (SDSS) is one of the largest, longest running, and most used astronomical surveys. Started in 1998 (Almeida et al. 2023), the SDSS provides all-sky, multi-epoch spectroscopy using telescopes in both hemispheres, providing data used to probe the emergence of chemical elements, reveal the inner mechanisms of stars, and investigate the origin of planets. The latest generation of the survey, SDSS-V, aims to conduct the first homogeneous survey using an optical, ultra-wide integral-field spectroscopic map of the interstellar gas, pioneering spectroscopic monitoring and revealing changes on both short and vast timescales (Kollmeier et al. 2017). SDSS data over all survey phases has been cited more than 650,000 times over 111,000 refereed papers (Almeida et al. 2023).

The SDSS has previously offered a simple web application for end users to access, explore, and investigate spectra and other data. However, the fifth generation of the survey now offers a large catalog of stars with complete stellar labels. With a larger set of spectral types explored (Majewski et al. 2017), and an even larger set of stellar labels (Casey et al., in prep), there is a need for a new tool which provides powerful exploration and visualization of large and vast datasets.

3 VisBoard Development

The fifth generation of the SDSS now provides complete stellar labels across multiple different pipelines via the Astra framework. This multi-dimensional parameter explorer, or by its less boring name, the *VisBoard*, is a in-development data webapp which will be hosted on SDSS servers, providing both public and proprietary SDSS data to its users for multi-dimensional data exploration via out-of-memory loading and computation using `vaex`.

Since the previous report, updates have been made which primarily improve the user experience (UX) of the software. The utility of the VisBoard has expanded to allow for exploration across a variety of viewing forms through an intuitive user experience. Much like a board of sticky notes, views can be added, removed, and resized at will by the user, whilst applying global filters across the selected dataset. Views themselves are dynamic, rerendering to filter out unnecessary points and saving zoom and pan information for a smoother UX.

In Section 3.1, we discuss this sticky note layout, and the new skyplot view and dynamic rerendering and filtering in Sections 3.1.1 and 3.1.2 respectively. In Section 3.2, we describe how the software supports multiple datasets as a proof-of-concept, and discuss future implementations of this functionality. Finally, in Section 3.3, we list other additional features added to the software since the last report.

3.1 Sticky Note Layout

The user can now directly add and remove multiple plotting views of the same time within their display, which can resized with a small handle in the bottom right, and moved via the top grey toolbar, as shown in Figure 1. Each plot also dynamically resizes to its containing "note", which can be seen in Figure 2.

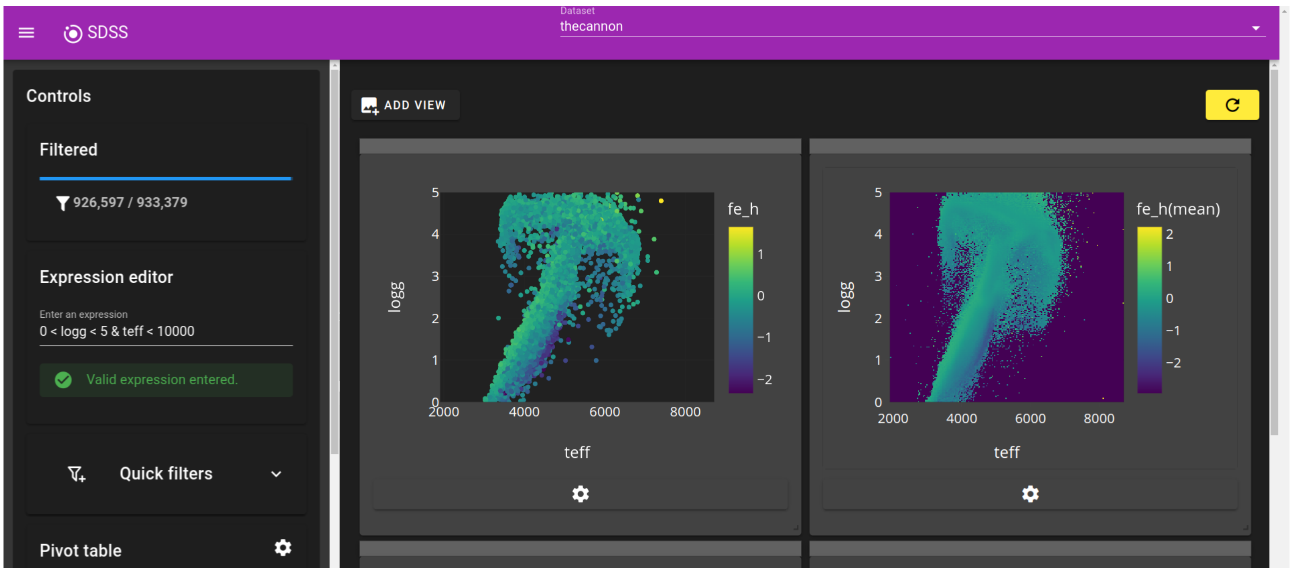
Within each view, a plot's individual properties can be changed, such as colorscale, the data to plot, and whether a given axis is flipped and/or logarithmic scale, as shown in Figure 3.

Each plot (excluding the *aggregation* view) now has a selection option, which allows the user to select certain bins or points, which will then be cross filtered to other open views.

On the scatter plot views, a user can right click on any given star to directly download or view the object's spectra in the browser via `jdaviz`. Currently, these lead to placeholders, but will eventually be replaced with browser links to the public SDSS data by the ID lookup.

3.1.1 Skyplot

New to this format is a scatter plot with a sky projection, herein referred to as a "*skyplot*". This view was heavily requested by astronomers after consultations and demos. The user can switch between



Dataset
thecannon

926,597 / 933,379

Expression editor
Enter an expression
0 < logg < 5 & teff < 10000

Valid expression entered.

Figure 1: A full screenshot of the application. Top: the Application bar, where the loaded dataset can be changed, currently viewing data from *The Cannon* (Ness et al. 2015). Left: the sidebar, containing filter controls, such as the expression editor. An expression to filter out values is currently applied. Right: the sticky note layout, showing the scatter and aggregation views, plotting $T_{\rm eff}$ and $\log g$.

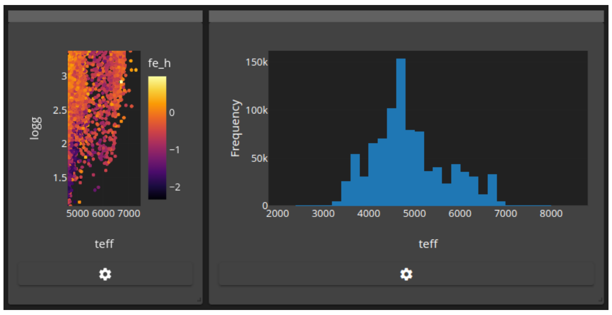


Figure 2: A demonstration of the dynamic resizing and draggable view “notes”. The plot objects dynamically resize to fit their container, and can be rearranged and resized by the user with their mouse.

plotting right ascension vs. declination (RA/DEC) or galactic longitude and latitude (l/b). Panning the view will accurately map the points according to the projection type, which can be set by the user. The settings menu and skyplot are shown in Figure 3.

3.1.2 Dynamic rerendering and filtering

Another new feature is the adaptive rerendering of Plotly’s `FigureWidget` objects, which consists of two parts:

1. Rerendering the plot with the same zoom and pan settings after a parameter change.
2. Filtering the dataset in the Scatter object views (*skyplot* + *scatter*) to the ranges of the zoomed plot in real time.

Previously, the plot would forcibly reset after any parameter change, such as changing the colorscale or binning type, which led to a disorienting UX that required the user to continuously redo their zoom/pan actions after parameter changes. Now, the plot continuously saves the visible x/y ranges, and calls upon them when plotting, only resetting it if the x/y data or plotting scale changes.

The other consequence of saving the relayout information is that we can directly filter the dataset to only include data inside the visible range, which allows the user to see more points after performing zoom or pan operations, bypassing the overplotting limitations of Plotly’s memory-heavy scatter points. This is similar in function to Holoview’s `DataShader` (Rudiger et al. 2020), although we do not rasterize the data prior to plotting.

3.2 Multiple Datasets

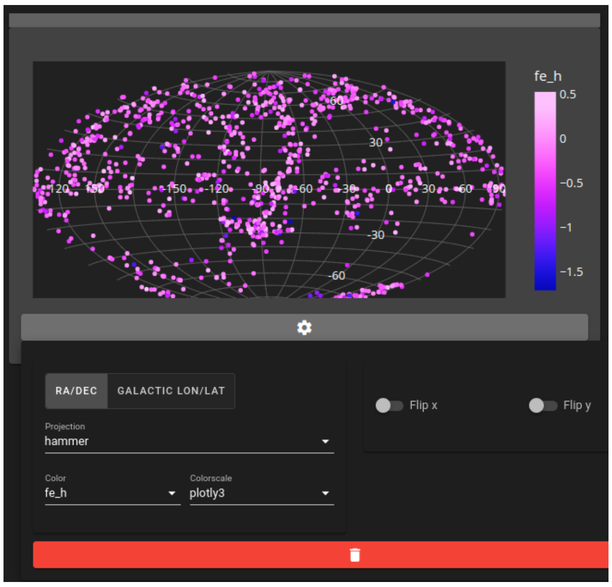
Within the top left of the application, the user can now directly select the Astra pipeline from which to access data from (see top of Figure 1).

Currently, this is implemented as a proof-of-concept feature, with the goal being to allow a user to select multiple pipeline datasets simultaneously, and apply unique filters and expressions directly between them or across all of them, and even their own given subsets.

The user’s subset of the chosen dataset can also be directly downloaded, based on their chosen filters.

3.3 Other features

- Expression editor updates, as shown in the left of Figure 1.
  - Added the ability to type expressions without brackets.



Projection: hammer
Color: fe_h
Colorscale: plotly3

Figure 3: Top: the *skyplot* view, showing right ascension (RA) against declination (DEC), and colored by metallicity [Fe/H] across a Hammer projection. Bottom: the settings menu of the skyplot view, where the coordinates, projection type, color values, colorscale, and axes flips can be changed by the user.

  - Added functionality to type 3-part expressions (i.e. $400 < x \leq 3000$)
  - Expression editor returns a more descriptive error message.
  - Expression editor catches expressions which reduce the dataset to a pointlessly low length.

- Added a Dark theme to the UI, matching the SDSS front-end framework `zora`.
- Added a Quick Filters menu for common filters (i.e. good results only)
- Added point-hover information to all views.
- Added selection filtering functionality to *histogram* and *scatter* plots.
- Added ascending/descending sorting options to the Table View card.
- (Partly) fixed a bug relating to CPU stride in `vaex`.
- Removed a defunct setting in Plotly.py's `imshow` plots.

4 Difficulties encountered

1. **`vaex` and `solara` documentation is often incomplete.**
   - Continuously, I found several API methods and hooks within the source code that were not apparent on the documentation.
   - This is common throughout open-source development, due to its decentralized nature, but I will also endeavour to ensure I properly document my work.
2. **Plotly's `FigureWidget` is very difficult to debug.**
   - Spanning across 3 codebases and over 60,000 lines of code, the `FigureWidget` has been difficult to debug.
   - The `FigureWidget` doesn't resize its height to its container, even though it's specified in CSS that it should.
     - I have no idea why, even after combing the codebase and testing different resize managers.
     - It also does not help that I am not fluent in Jupyter widget libraries, Typescript, or Javascript.
   - Solving this problem is out of scope for the project's timescale, so the views are height-locked until this is fixed.

5 Future plans

1. **Add dynamic filtering to the *skyplot* view.**
   - Due to the difference in relayout callback information, it was not simply plug and play to implement the routine.
   - It wasn't finished prior to writing this report, but the logic is sorted already and just needs to be implemented.
2. **Develop a `Singularity` container for shared development.**
   - The Data Visualization group within the SDSS Collaboration will soon be granted access to a virtual machine (VM) for testing and developing applications.
   - Members of the collaboration are interested in providing feedback and testing the application, and have suggested using `Singularity` (Kurtzer 2018) as the portable app-container for the development workflow.
3. **Expand upon dataset functionality.**
   - As described in Section 3.2, I plan to expand upon the dataset functionality.
   - The exact implementation of this feature and method depends on how data will be stored across different access methods, which is yet to be decided.
     - i.e. will it be a single file containing all the data? Will it be constantly updated? Or will there be multiple files that must be combined?

4. **Expand upon vaex methods.**
    - vaex's features to provide lazy, out-of-memory computations is of interest for features in the exploration application.
    - I plan to add features that allow the user to analyse different subsets based on a groupby routine, and create virtual columns that are based on lazy computations on data from other columns (see vaex's documentation for further reading).

References

Almeida, Andrés et al. (Aug. 1, 2023). "The Eighteenth Data Release of the Sloan Digital Sky Surveys: Targeting and First Spectra from SDSS-V". In: *The Astrophysical Journal Supplement Series* 267. ADS Bibcode: 2023ApJS..267...44A, p. 44. ISSN: 0067-0049. DOI: 10.3847/1538-4365/acda98. URL: https://ui.adsabs.harvard.edu/abs/2023ApJS..267...44A (visited on 08/22/2023).

Kollmeier, Juna A. et al. (Nov. 2017). "SDSS-V: Pioneering Panoptic Spectroscopy". In: *arXiv e-prints*. _eprint: 1711.03234, arXiv:1711.03234. DOI: 10.48550/arXiv.1711.03234.

Kurtzer, Gregory M et. al. (July 2018). *Singularity 2.5.2 - Linux application and environment containers for science*. Version 2.5.2. DOI: 10.5281/zenodo.1308868. URL: https://doi.org/10.5281/zenodo.1308868.

Majewski, Steven R. et al. (Sept. 1, 2017). "The Apache Point Observatory Galactic Evolution Experiment (APOGEE)". In: *The Astronomical Journal* 154. ADS Bibcode: 2017AJ....154...94M, p. 94. ISSN: 0004-6256. DOI: 10.3847/1538-3881/aa784d. URL: https://ui.adsabs.harvard.edu/abs/2017AJ....154...94M (visited on 09/20/2023).

Ness, M. et al. (July 2015). "The Cannon: A data-driven approach to Stellar Label Determination". In: ApJ 808.1, 16, p. 16. DOI: 10.1088/0004-637X/808/1/16. arXiv: 1501.07604 [astro-ph.SR].

Rudiger, Philipp et al. (June 2020). *holoviz/holoviews: Version 1.13.3*. Version v1.13.3. DOI: 10.5281/zenodo.3904606. URL: https://doi.org/10.5281/zenodo.3904606.