A Future for Data Exploration: a proof-of-concept data explorer

Riley Thai, Andy Casey (Supervisor)

February 15, 2024

Abstract

We developed a new, multi-dimensional parameter explorer web application, named the *Visboard*, for online data exploration of the upcoming data release of the Sloan Digital Sky Survey (SDSS). We also developed default loaders for `Astropy`'s `specutils` subpackage, allowing for astronomers to load the new datatypes in the upcoming SDSS data releases to be loaded directly as Python objects for manipulation with other libraries, such spectrum visualization tools like `jdaviz`. In this report, we showcase the key features of these software applications, the design decisions made, and discuss the future plans for the development of the *Visboard*.

1 Introduction

Astronomers survey the night sky and collect data from a huge number of objects, from the stellar parameters and abundances of stars, to the accretion rate of black holes. By providing large catalogs of objects, astronomers and astrophysicists alike can explore these datasets to test their hypotheses, validate the results of others, and develop our understanding of the universe at large.

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. SDSS data over all survey phases has been cited more than 650,000 times in over 111,000 refereed papers (Almeida et al. 2023).

The fifth generation of the survey, SDSS-V, The latest generation of the survey, SDSS-V, aims to conduct the first homogeneous survey using an optical and infrared (IR), 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).

The survey's key scientific goals are bundled into three different programs.

The Milky Way Mapper (MWM) will employ both optical and near-IR spectroscopy across over 5 million stellar objects, providing a dense, contiguous map of the sky with a high-dimensional parameter space, including stellar mass, age, chemical composition, rotation, and internal structure. The MWM aims to recreate the deep history of the Milky Way Galaxy through quantitive tests of physics-based galactic formation models against its large, high-dimensional catalog (Kollmeier et al. 2017).

The Black Hole Mapper (BHM) studies quasars and their supermassive black holes (SMBHs) at the center of galaxies to trace the evolution of BHs across cosmic timescales (Kollmeier et al. 2017). Through optical spectroscopy, the BHM will provide mass measures and multi-wavelength spectral energy distributions for a large sample of over 300,000 quasars, developing insight into the co-evolution of SMBHs with their host galaxies.

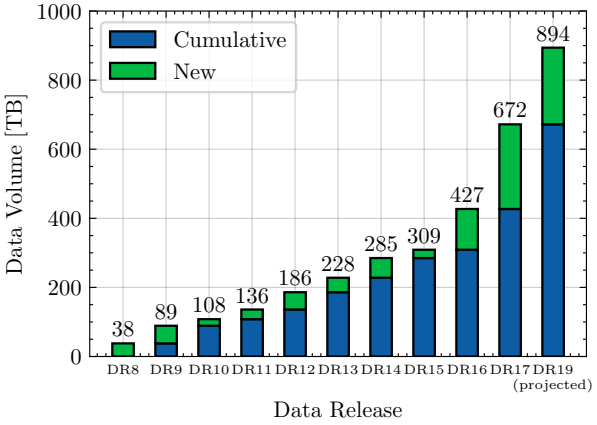

Figure 1: A histogram of the cumulative data volume of each Data Release of the SDSS since Data Release 8 (DR8). The next Data Release, DR19, is projected to have 1.3x the amount of data in Data Release 17, with the cumulative data volume totalling to possibly over 900 terabytes.

The Local Volume Mapper (LVM) aims to study the self-regulating processes of galactic formation by creating global interstellar medium (ISM) maps of Local Group galaxies and nebulae through high-resolution telescopes and spectrographs (Kollmeier et al. 2017). The LVM will return data aimed at researching star formation and the physics of the ISM through revealing individual star formation knots and the shock networks of filamentary structures across local nebulae and galaxies.

With the large number of new scientific goals, the total data volume of the survey has increased dramatically, as shown in Figure 1. The SDSS has previously offered a simple web application for end users

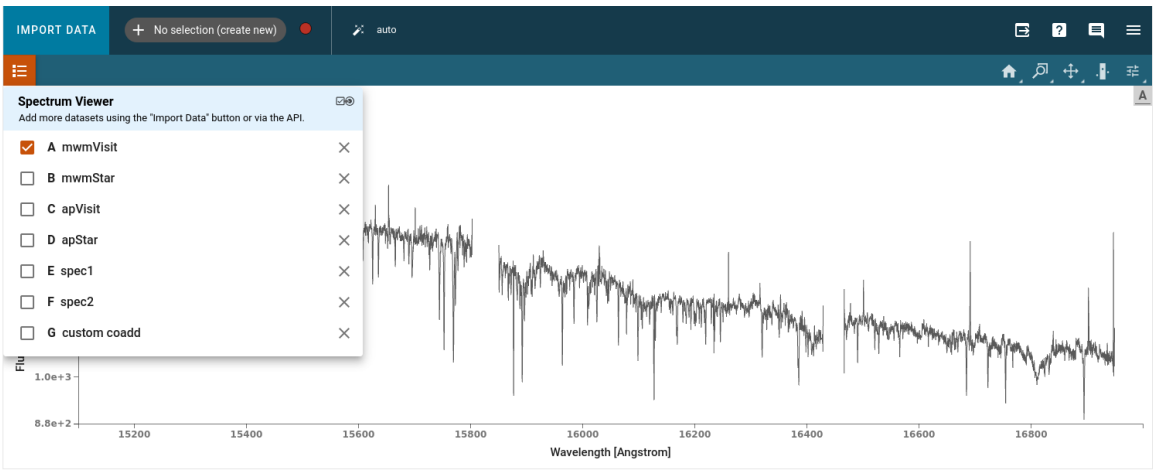

Figure 2: Spectrum data from a BOSS spectrograph, as shown within `jdaviz`'s spectrum viewer, *Specviz*. Note that other spectra are also loaded, and can be toggled by the checkboxes in the open dropdown menu.

to access, explore, and investigate spectra and data catalogs. However, since the fifth generation of the survey is offering a massive, high-dimensional catalog of several different types of galactic objects, there is a need for a new tool which provides powerful exploration and visualization of these new, larger datasets. This tool aims provide visualization of SDSS data, without requiring the user to download over several terabytes of data for their research. Specifically, the collaboration requires a web application with a low server load, powered by new libraries which serve data "out-of-core", meaning that they only load necessary chunks of the data, as opposed to the entire dataset at once.

In this report, we detail the development work undertaken and software features of a multi-dimensional, out-of-core parameter explorer for SDSS data, alongside the implementation of automatic loaders for SDSS datatypes within the Python library `specutils`, providing a set of widely applicable software tools for handling astronomical spectra under the highly impactful `Astropy` project. In Section 2, we describe the features implemented into `specutils` and the utility they provide. In Section 3, we showcase and outline the parameter axplorer and its features, describing the design decisions and the key software functionality. Finally, we outline our future plans in Section 4, and write our conclusions in Section 5.

2 `specutils` Default Loaders

The Python library `specutils` is package which provides a Python object representation of spectra and basic operation tools (Earl et al. 2023). The package is part of the `Astropy` project, which aims to provide a centralized and widely applicable set of tools for handling astronomical data (Astropy Collaboration et al. 2022). Of note is its integration with spectrum viewers and analysis tools, such as the `jdaviz` package, which provides data viewers and analysis plugins for interactive applications (Developers et al. 2023).

Of use to the SDSS collaboration and the wider public would be the feature to automatically load the new and updated SDSS datatypes with `specutils` spectrum representations, namely the `Spectrum1D` and `SpectrumList` objects.

As such, the software I developed provides default I/O reading functions for both the new and updated datatypes within the SDSS-V dataset. There were changes to the location of wavelength and spectral flux properties within the `apStar`, `apVisit`, and SDSS `spec` datatypes. It can handle loading spectra files both locally and via a direct web link directly as `Spectrum1D` or `SpectrumList` objects.

The handlers are built-in as the sub-method to each `Spectrum` class, by specifying a filepath, a user can also choose to read a single or all spectrum from a given SDSS data file.

Most importantly, other libraries which use the `Spectrum` objects to load the new SDSS-V datatypes, such as the `jdaviz` spectrum viewer, can now read SDSS-V datatypes. Several loaded spectrum objects from SDSS-V spectra are shown in Figure 2.

The pull request to merge this functionality into the `specutils` package is currently under review, and can be viewed here for further reading.

3 Parameter Explorer

The fifth generation of the SDSS now provides complete stellar labels across multiple different pipelines via the Astra framework. The application we developed, named the *Visboard*, is a in-development data webapp which will be hosted on SDSS servers and provide both public and proprietary SDSS data to its users for multi-dimensional data exploration.

The *Visboard* provides data 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

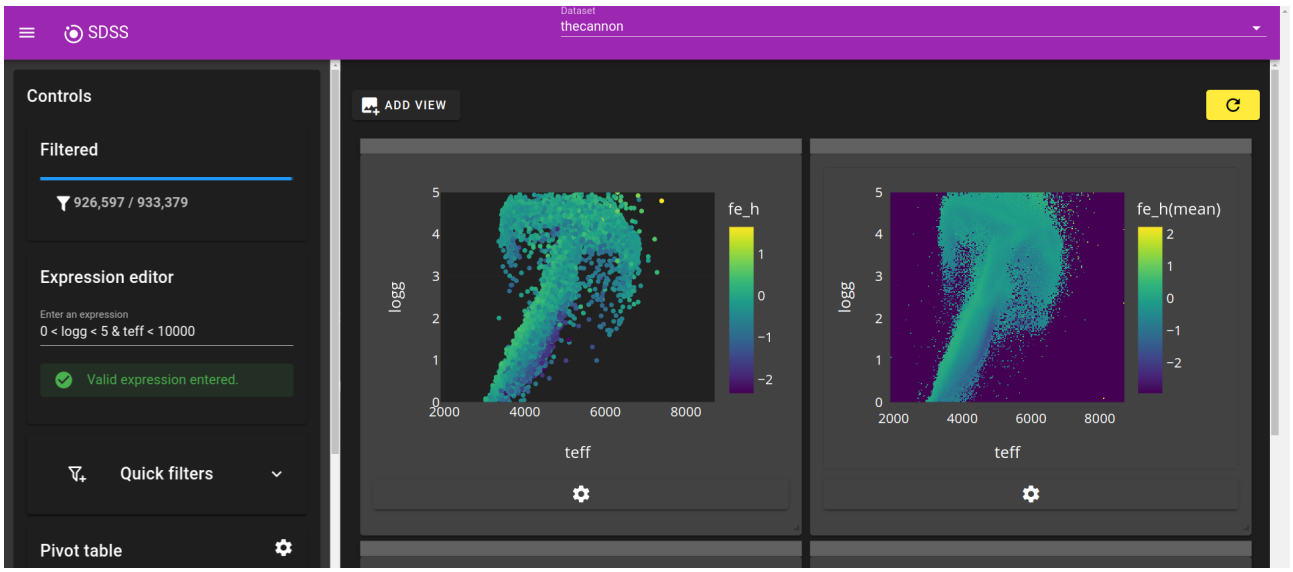

Figure 3: A full screenshot of the application. Top: the application's toolbar, where the loaded dataset can be changed. Currently viewing data from *The Cannon* (Ness et al. 2015). Left: the sidebar, containing filter controls, such as the expression editor. An expression to filter out values is currently applied. Right: the sticky note layout, showing the scatter and aggregation views, plotting T_{eff} and $\log g$.

dataset. Views themselves are dynamic, rerendering to filter out unnecessary points and saving zoom and pan information for a smoother user experience. Users can choose between different SDSS datasets and both unique and common filters to reduce the dataset to their desired scientific subset.

In Section 3.1, we discuss the software choices for the library and outline our reasoning. Throughout Section 3.2, we discuss the key features of the application. In Section 3.2.1, we discuss the application's sticky note layout, the plotting views, such as the aggregated (3.2.1.1) and skyplot plots (3.2.1.2), and the dynamic rerendering across these views in Section 3.2.1.3. In Section 3.2.2, we discuss the application's cross-filtering functionality across all views and the user-facing dataset filtering, called Expressions. In Section 3.2.3, we describe how the software supports multiple datasets and discuss future implementations of this functionality.

3.1 Technology

The *Visboard* utilizes various Python packages, as Python has become the premier language for astronomical analysis and data science. Three packages were core to the application's functionality.

1. `solara` – a Python, React-like web application framework.
2. `vaex` – a data science focused package for handling very large datasets out-of-core (Breddels and Veljanoski 2018).
3. `Plotly.py` – a highly interactive graphing and plotting library (Plotly Technologies Inc. 2015).

The choice for using `solara` was experimental, but necessary. Released in 2022, the `solara` library builds on existing technology, such as Vuetify components and the `ipywidgets` system. Being written in Python, it allows for high speed development workflows through easier integration with data science packages already written in Python, such as `vaex`.

`vaex` specifically is the fundamental core of the application's backend data processing, as it provides aggregated statistics and out-of-core data handling on over 1 billion rows of data per second. `vaex` was specifically developed to handle a new era of big data astronomy, such as Gaia and the upcoming LVM data (Breddels and Veljanoski 2018). Futhermore, the data is encoded in the Apache Parquet format, which uses the Apache columnar data format for more efficient in-memory processing and data interchange (Vohra 2016).

Finally, `Plotly.py` was chosen as the visualization library due to its highly interactive components, which allow for distinct user hover information and interactive selections of the dataset. Several other alternatives were explored, such as `Bokeh` and `Matplotlib`. These packages were not chosen for visualization, as they were either too difficult to implement as an interactive element in `solara` (as with `Bokeh`), or they did not support an interactive interface natively.

The two other packages used were `numpy` (Harris et al. 2020) and `xarray` (Hoyer and Hamman 2017), which provided several quantitative data calculation methods.

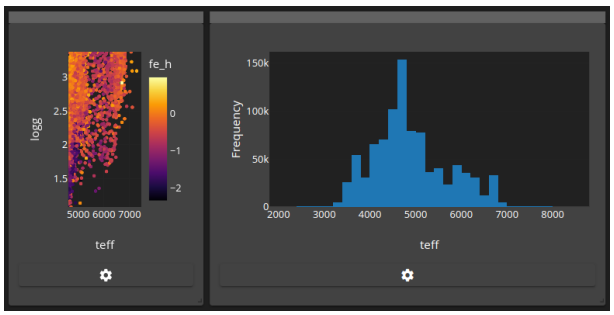

Figure 4: A demonstration of the dynamic resizing and draggable view "notes". The plot objects dynamically resize to fit their container, and can be rearranged and resized by the user with their mouse.

3.2 Application Features

3.2.1 Layout and Views

The user can 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 3. Each plot also dynamically resizes to its containing "note", which can be seen in Figure 4.

There are 5 different views with can be added to the layout – histogram, scatter, aggregated, skyplot, and table views.

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 8.

Each view has a selection option, which allows the user to select certain bins, points, or rows, 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 the spectrum of that object, or view the object's spectra via `jdaviz`. Currently, these lead to placeholders, but will eventually be replaced with browser links to the public SDSS data by the ID lookup.

3.2.1.1 Aggregated

The aggregated view utilizes an image render based on Plotly's `imshow` to provide a high detail 3 dimensional view. By utilizing `vaex`'s aggregated statistics over n-dimensional grids, which allows for complex multi-dimensional plotting, the view bypasses the pitfalls of scatter plots, such as overplotting or underrepresentation through downsampling. This view is shown in Figure 5.

The view can show the mean, minimum, maximum, or mode of any cell on the grid, and the size of the grid can be adjusted.

3.2.1.2 Skyplot

The skyplot is a scatter plot with a sky projection, herein referred to as a "*skyplot*". This view was heavily requested by astronomers after consultations and demonstrations. The user can switch between 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 8

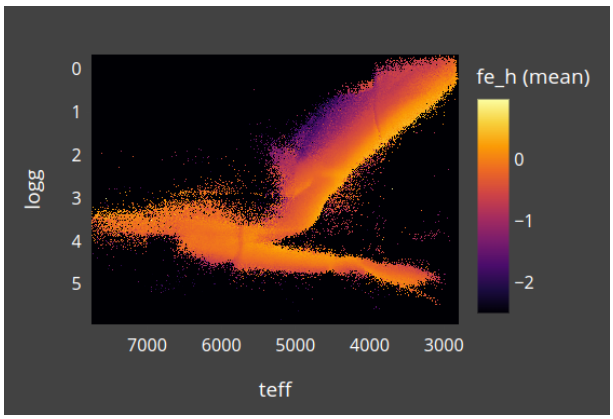

Figure 5: A display showing the aggregated view, plotting the ASPCAP catalog objects' effective temperature $T_{\rm eff}$ vs. surface gravity $\log g$, colored by the mean metallicity [Fe/H]. The axes are inverted.

3.2.1.3 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.

In previous versions, the plot would forcibly reset after any parameter change, such as changing the colorscale or binning type, which led to a disorienting user experience 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 it allows for us to 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 allows for us to have an effectively infinite zoom with complete hover information, allowing for precise dataset exploration across bot the scatter and skyplot views. This is similar in function to Holoview's `DataShader` (Rudiger et al. 2020), although we do not rasterize the data prior to plotting.

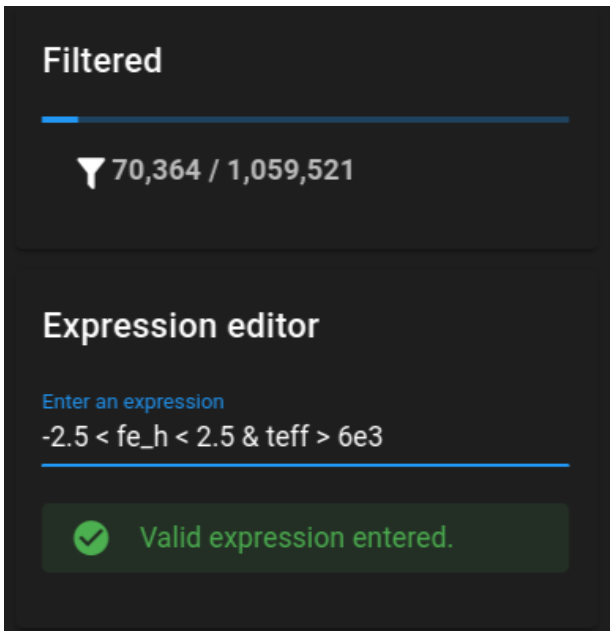

Figure 6: A truncated view of the sidebar of the explorer, with an Expression applied to the dataset. Top: The number of rows in the current filtered dataset. Bottom: The expression editor.

3.2.2 Cross-filtering

Within the application's sidebar on the left hand side (see Figure 3), one can access dataset filtering controls, such as an Expression Editor, Quick Filters, and a Pivot Table.

The Expression Editor allows the user to leverage `vaex`'s high-speed computations to filter the dataset. This can be combined with other filtering methods, the pivot table and/or selection, to directly select a complex subset of data. A truncated view containing the Expression Editor is shown in Figure 6.

The Quick Filter menu contains checkboxes to quickly apply common filters. This was a feature suggested by astronomers after demonstrations within SDSS telecons. Currently, it is implemented with the basic functionality to limit the dataset to high SNR subsets (SNR > 50), or only include non-flagged results. The functionality can be expanded in future to potentially hold user filters, or alternatively limit to different cartons.

The Pivot Table also allows a user to filter the dataset based on a specific result in a distinct UI, powered by `vaex`'s `group-by` objects. Users can click on a given set of the data to filter the dataset to that row, column, or cell of the pivot table. The user's subset of the chosen dataset can also be directly downloaded, based on their chosen filters. The pivot table is shown in Figure 7.

3.2.3 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 3).

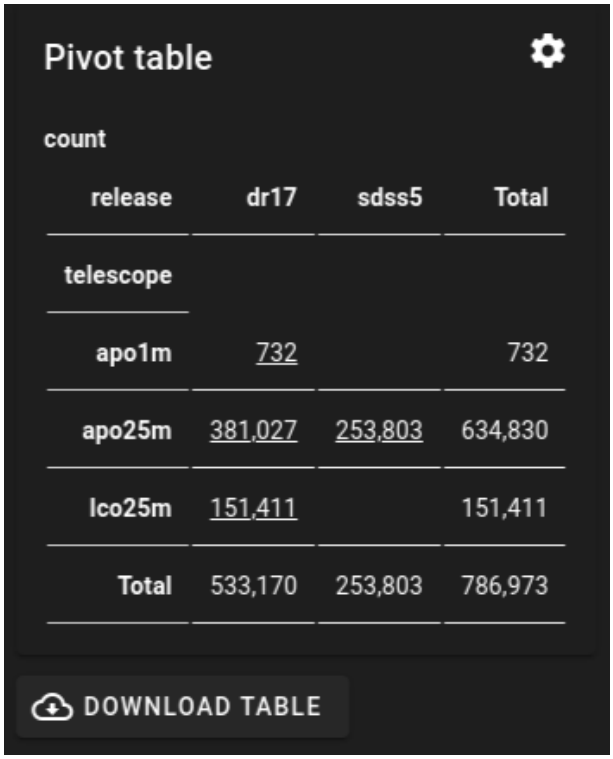

Figure 7: A view of the pivot table, showcasing release vertically, and telescope horizontally. Users can click on a given set of the data to filter the dataset down to that row, column, or cell of the pivot table. Further specifications can be added by the cog icon in the top right.

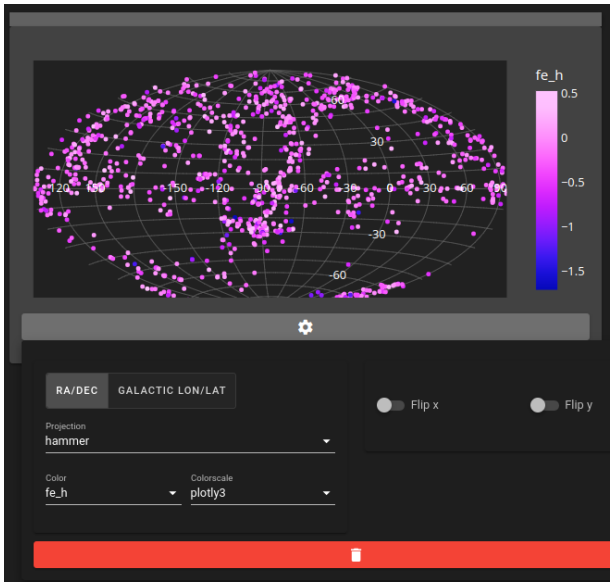

Figure 8: Top: the *skyplot* view, showing right ascension (RA) against declination (DEC), and colored by metallicity [Fe/H] across a Hammer projection. Bottom: the settings menu of the skyplot view, where the coordinates, projection type, color values, colorscale, and axes flips can be changed by the user.

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 exact implementation of this feature and method depends on how data will be stored across different access methods, which is yet to be decided as it depends on the data volume organization and pipeline output frequency for the working dataset and the public Data Release.

As shown previously, the user's subset of the chosen dataset can also be directly downloaded, based on their chosen filters.

4 Future Plans

While the majority of core development features are completed, there is still work required to expand its functionality, polish the application, and improve its performance.

Since the user experience is of vital importance, we seek to present this application to the collaboration several more times before the launch of Data Release 19 in late 2024 for feedback. Their input is vital for understanding the scientific workflow and the necessary tools and integrations. Currently, suggested ideas have included the multiple dataset functionality, saving and loading different filters, quick carton-based filtering, and accessibility improvements, such as font sizes and colorblind-accessible colorscale palettes.

To allow for users to understand the application thoroughly, we plan to develop a deployable container on an SDSS server, so users can use the application in their own time for their own science or testing. 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.

Some other key features we are looking to implement are the loading of several datasets simultaneously from different pipelines, such as loading an ASPCAP and ApogeeNET dataset simultaneously, which will help with pipeline performance assessment. We also plan to expand upon `vaex`'s features to provide lazy, out-of-memory computations, such as virtual columns based on user calculations.

The performance of the application has decreased since earlier versions. This is due to the number of components, callbacks, and hooks increasing dramatically without proper state co-location and memoization, which can prevent unnecessary rerenders. The codebase must be revised to ensure that the application doesn't perform these rerenders and other variable checks, especially with regards to the plotting code.

There are also still numerous bugs, which will be resolved in course through the continued development of the application. Nonetheless, the plan is for *Visboard* to be deployed with the web portal for Data Release 19 in late 2024.

5 Conclusion

In this report, we have discussed the development and design of a new web application for the fifth generation of the Sloan Digital Sky Survey. In Section 2, we described the features implemented into `specutils` and the utility they provide. In Section 3, we showcased and outlined the parameter axplorer and its features, describing the design decisions and the key software functionality. Finally, we outlined our future plans in Section 4.

The ultimate challenge for modern data web applications in this new era of big data astronomy has been the lack of proper software tools which can support the new, larger data volumes of Gaia, eROSITA, and the upcoming LVM of SDSS-V. However, new Python libraries which support out-of-core, aggregated statistics are emerging, alongside modern data visualization libraries with highly interactive elements. Coupled with efficient web technologies, data can now be served to users directly without the need for them to download everything, allowing for astronomers and other big data professionals alike to conduct high-level data exploration of datasets too large for in-memory analysis.

This technology is expected to be deployed for SDSS's Data Release 19, which is expected to launch in late 2024.

Special Thanks

I thank Andy Casey, my supervisor, for his assistance with this project. Specifically, I thank him for permitting me to work fully remotely with minimal overhead supervision.

I would like to thank Brian Cherinka and Joel Brownstein of the SDSS Collaboration/University of Baltimore for providing me with this opportunity to work on this project.

I also extend my thanks to the Data Visualization Group and Astra Group of the working SDSS Collaboration, for their feedback and support through the development of this application.

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).

Astropy Collaboration et al. (Aug. 2022). "The Astropy Project: Sustaining and Growing a Community-oriented Open-source Project and the Latest Major Release (v5.0) of the Core Package". In: ApJ 935.2, 167, p. 167. DOI: 10.3847/1538-4357/ac7c74. arXiv: 2206.14220 [astro-ph.IM].

Breddels, Maarten A. and Jovan Veljanoski (Oct. 2018). "Vaex: big data exploration in the era of Gaia". In: A&A 618, A13, A13. DOI: 10.1051/0004-6361/201732493. arXiv: 1801.02638 [astro-ph.IM].

Developers, JDADF et al. (Dec. 2023). *Jdaviz*. Version v3.8.1. DOI: 10.5281/zenodo.10420627. URL: https://doi.org/10.5281/zenodo.10420627.

Earl, Nicholas et al. (Oct. 2023). *astropy/specutils: v1.12.0*. Version v1.12.0. DOI: 10.5281/zenodo.10016569. URL: https://doi.org/10.5281/zenodo.10016569.

Harris, Charles R. et al. (Sept. 2020). "Array programming with NumPy". In: *Nature* 585.7825, pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: https://doi.org/10.1038/s41586-020-2649-2.

Hoyer, S. and J. Hamman (2017). "xarray: N-D labeled arrays and datasets in Python". In: *Journal of Open Research Software* 5.1. DOI: 10.5334/jors.148. URL: http://doi.org/10.5334/jors.148.

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.

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].

Plotly Technologies Inc. (2015). *Collaborative data science*. URL: https://plot.ly.

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.

Vohra, Deepak (2016). "Apache Parquet". In: *Practical Hadoop Ecosystem: A Definitive Guide to Hadoop-Related Frameworks and Tools*. Berkeley, CA: Apress, pp. 325–335. ISBN: 978-1-4842-2199-0. DOI: 10.1007/978-1-4842-2199-0_8. URL: https://doi.org/10.1007/978-1-4842-2199-0_8.