Python API for visualisation of sets

Extended Abstract

Nathan John Crowley
118429092

Department of Computer Science
BSc Computer Science
Dr Rosane Minghim
University College Cork

March 2022

Abstract

The project involves creating a Python API/ library to be used by students or academic professionals with the aim of visualising sets, using InteractiVenn's approach to Set Visualisation, to allow a greater understanding of given sets, and their unions and intersections. The Python API developed should allow users to input up to six sets of data to be visualised as an interactive Venn Diagram with functionality that:

1) Allow users to visualise the union and intersections of the given complex input sets.
2) Allow users to hover over set unions and present the elements of a given set.

The developed Python API should take into account that the users may have little or no technical skills, this should direct the design towards being as easy as possible to use for the target audience. The documentation of the Python API must have clear and concise information on its functionality, along with concrete examples displaying to the user the operations that can be performed.

The project includes running and evaluating the developed Python API by itself as well as against an established API, UpSet – Visualising Intersecting Sets, to highlight the advantages and disadvantages of each API given a number of examples.

Introduction

The project's goal is to create a Python Application Programming Interface (API) to allow visualisation and exploration of sets and Venn-diagrams for students or academic professionals, to improve the understanding of the correspondences between sets, such as unions and intersections. Python API's are an importable, reusable section of code hosted on a repository, eg.Python Package Index (PyPI) [1], that can be included in a user's program or project. The Python API should handle any downloads of necessary dependencies to improve user experience.

The Python API created will have a target audience of Scientific professionals and/or students who regularly require set visualisation. In addition the developed Python API should take into account that the users may have little technical skills. The Python API created will be evaluated against existing set visualisation technology, UpSet https://upset.app/. The created Python API and UpSet will be evaluated side by side with a number of examples, with the target of examining each technologies advantages, disadvantages and the appropriate use case.

API documentation containing the technical content will be provided. Providing users with knowledge of the available methods and functionality of the API. Included will be information regarding the importing of the API from PyPI, along with input examples for the API and the expected output. Documentation for the Python API created can be found at:
https://github.com/NathanCrowley/FinalYearProject/blob/main/ReadME.md.

Technologies Used

Throughout the development lifecycle of the Python API project numerous technologies were used for a wide range of necessary operations such as: code generation, hosting, testing, documentation, etc. The below paragraphs will explain the use and benefits of each technology used.

Python programming language [2] used as the base language for the API created. Python was the programming language assigned to the project in the project description. Python is beneficial to the project as it is accessible to all technical levels of users, allowing both technically experienced together with users who have little or no technical skills to interact with the API. Matplotlib: Visualisation with Python [3], a plotting library for the Python programming language, allows the Python API created to produce the required outputs for the given user inputs of sets to be represented as a Venn Diagram. PyPi: Python Package Index [1], offers developers the ability to host their Python API or Python libraries. The Python API is uploaded to PyPi by the developer. The user of the project can import the Python API using the python Pip package installer for Python [4] or from the PyPi website. Test PyPI [5] is a separate instance of PyPi. Test PyPi used by the developer to privately upload project versions and test importation prior to releasing a given version on PyPi.

Github, Inc [6] hosts the project's source code throughout the development of the Python API. Github along with the open-source tool Git [7] allowed for continuous development and version control for the duration of the Python API development.

LaTeX document preparation system [8] is the recommended document generation tool for scientific reports, such as the Final Project Report deliverable.

Project To Date: 9th March 2022

Venn diagrams [9] are a diagram style that uses overlapping circles, with each representing a set of elements. Allowing a reader to efficiently examine relationships such as: set intersections and set unions.

Base directory "/FinalYearProject" contains the entire contents of the project:

1. Deliverables: the 'Deliverable Requests' listed on University College Cork's final year project website [10].
2. FYP_Drive: meetings notes and files shared between myself and my supervisor, Rosane Minghim, in the Microsoft Teams team 'FYP : InteractiVenn as an API'.
3. Report_Tips: any helpful files shared from the supervisor with advice on report generation and format.
4. Research: working directory of all experimental research completed over the final year project's lifecycle, with sub-directories:
    4.1. API_research: research conducted on Python API examples and implementation.
    4.2. Library_resarch: testing and implementation of simple Python API's.
    4.3. PDF: back-up PDF versions of the primary deliverable request, the Final Project Report.
    4.4. Upset: interacting with the primary technology for evaluation, UpSet API [11].
    4.5. VennDiagram_research: research on simpler examples of using Python to create and output venn diagrams on example input sets.
        4.5.1. VennDiagram_research/VennDiagramLibrary: contains the source code for the 'drawVenn' API created.
5. TestAPI: working test directory containing a jupyter notebook [12] testing file, 'Testing.ipynb' used to import, from PyPI, the Python API created to test the outputs generated and evaluate them against expected output.
6. ReadMe.md: markdown file containing documentation of instructions of how to take advantage of the functionality of the API.

The API working directory is found at 'FinalYearProject/Research/VennDiagram_research/VennDiagramLibrary/drawVennDiagram'. Source code for the Python API created is found at './__init__.py'. The API consists of a section of imports and default values for the API's variables. The API takes advantage of Python Object Oriented Principles [13] as the API is represented as a Python class object with methods that provide functionality to modify and create venn diagrams. Dependencies needed for the API functionality can

be found at ‘./requirements.txt’, these will be automatically installed, as stated in the ‘setup.py’ file, once the API is installed from PyPI.

To date, 09/03/2022, users can import the API using the command ‘pip install drawVennDiagram’. Once imported the user creates the necessary Python built-in set objects, along with stating each set label using a tuple. The set objects along with set labels can be passed to the API using the ‘drawVenn() function’. The user can then use the built-in Python print() function, that has been overwritten to output the expected venn diagram. Current state of the project is limited to two or three set Venn Diagrams.

Example code for installation, set creation, label selection, creation of drawVenn object and outputting through the python print function.

```
pip install drawVennDiagram
set1 = set([1,2,3])
set2 = set([3,4,5])
labels = (‘A’,’B’)
vennd1 = drawVenn([set1,set2], labels)
print(vend1)
```

Next Steps for Project

The future project state should extend the development from the previous chapter. The goals of future development include:

1. Adding functionality to the API to allow the user to position their mouse over the generated venn diagram output and if a venn diagram section is selected, output the elements in this given section of the venn diagram.
2. Current state of the API is limited to inputs of two or three sets, future states of the project should allow users to input up to six sets to the API.

Documentation must be created describing the resources, functionality available. that the user can implement. The documentation created should be accessible to users of all levels of technical experience. Concrete working examples will guide the user on how to interact with the API.

The Python API created must be evaluated in comparison to an existing set visualisation technology, UpSet. The evaluation should display the key benefits and disadvantages of each technology, through concrete examples given to both. Continued development of the Final Project Report deliverable, showcasing all development and evaluation completed for the project's lifecycle.

Conclusion

This report and research would not have been possible without the guidance and support of my supervisor Dr Rosane Minghim. Dr.Minghim's supervision has been critical from the beginning of the project to the conclusive report. Her knowledge and management showcased in bi-weekly meetings, manifested in meeting reports as well as a continuous logbook of the project's progress, helped keep the project and myself on track to achieve the goals set for the project. The project interested me based on my API experience gained during the third year Work Placement module completed in 2021.

Appendix

[1] PyPi, Python Package Index (PyPI) is a repository of software for the Python programming language.Developed and maintained by the Python community, for the Python community. (2022), https://pypi.org/

[2] Python programming language (Python), https://www.python.org/.

[3] Matplotlib Visualisation with Python, https://matplotlib.org/.

[4] Pip package installer for Python, https://pypi.org/project/pip/.

[5] Test PyPI, https://test.pypi.org/.

[6] Github, Inc, https://github.com/.

[7] Git open source tool, https://git-scm.com/.

[8] LaTeX document preparation system, https://www.latex-project.org/.

[9] Venn Diagram, popularised by John Venn, https://en.wikipedia.org/wiki/Venn_diagram. (see https://www.lucidchart.com/pages/tutorial/venn-diagram for Venn diagram terminology)

[10] University College Cork's final year project website deliverables, https://project.cs.ucc.ie/deliverable/index.

[11] UpSet API, https://upset.app/.

[12] Jupyter notebook for API testing, https://jupyter.org/.

[13] Python Object Oriented Principles, https://medium.com/swlh/the-4-pillars-of-oop-in-python-9daaca4c0d13#:~:text=Abstraction%2C%20Encapsulation%2C%20Inheritance%2C%20Polymorphism,%E2%80%9D%20of%20Object%2DOriented%20Programming.