Development Report

WEATHER DATA VISUALISATION

Angelos Athanasakos / 40086764 | Honors Project | 12/10/2018

Words: 2600

Contents

1. Introduction

Data visualization is without debt one of the most important aspects in the Information Technology field and in the recent years has been gaining more momentum as the acquisition of big data sets have become widely available. Scientists have been looking a way to visualize any kind of data using the appropriate method which critically visualises the key elements that must be identified, therefore it is crucially important to consider which visual methods are available to the scientists. These Data Visualizations can be enhanced when combined with an element of interactivity within an easy to use web application that is responsive and accessible.

When a layer of interactivity is added, as with anything that user interacts in real, it gives a different perspective and creates a personal touch between the system and the user. For example, in the context of Data Visualization it would be beneficial to the user to go more in depth to a large data set or even more load simultaneously different data sets, compare and create correlations between data and analyze a specific focus area which would be much more difficult to do so before.

Reflecting to the possibility of creating an Interactive Web Application, the creation of a Web Application was considered using weather data sets from the Met Office and investigate technologies that would be technically achievable to integrate a degree of interactivity between the user and the application. It would consist of initially as a single page application, with an interactive map which would contain three weather station contain historical weather data from 1957 until 2017. The data which would be visualized are Sunshine hours, Minimum and Max Temperature, Precipitation and number of Air Frost Days. The user would be able to visualize each weather station, get some key weather-related information for that specific weather station and then he would be able to interact in a stimulating manner with the visualizations.

The final deliverable will be a fully functional web application which will be fully documented, follow the agile development practice and meet the minimum requirements as proposed in the learning contract but also a provision for future work, such as UX evaluation through task completion to understand whether it is user friendly or accessible but also technical aspects such as a weather data comparison, allow the user to upload each own data set or even the development to a fully functional CMS.

2. Design

The technical specifications that follow were carefully investigated and considered under the literature review. There were a lot of different technologies that could be used for Data Visualization. That ranged from a lot of different Frameworks consisting of R, Python and JavaScript. There were also platforms which were specifically designed for Data Visualization but were very limited in terms of features, expandability and user interactivity. Therefore, the only one was the most suited for building an Interactive Web Application that covered my requirements was Python.

Python is versatile programming language that can be used in many different scenarios. It is easy to use, has a clean syntax and has a huge variety of frameworks and dependencies for any type of requirements. In relation to my project Python Dash is created by Plotly and is used for building analytical web applications (Plotly, 2018). It is a library which is based on Python Flask and React frameworks. It is a relatively new technology which was released in 2017 and is specifically designed for building Interactive Web Application for data visualization of any type. It is completely open source which comes to as an advantage when considering expandability and prospects. Dash is designed for creating the interactive graphs which includes filtering, zooming as well as data isolation. These standard features and greatly improves the development speed of relevant project, as these features are challenging to implement.

Since it is a relatively new framework the challenges were many along the development process. The documentation was limited and the online resources very difficult to find. Therefore, the only options were to consult the limited official documentation and similar web applications that were previously made to get a perspective of the structure of a professionally made project. A good example would be refereeing to the official website and the officially produced application that demonstrate the technology.

The combination of additional frameworks such as, Pandas and Bootstrap 4 were essential to accomplish the realization of the application. More importantly Pandas framework is specifically integrated in the data manipulation of the data sets. It is worth mentioning that it is widely used as a tool in Data Science and Data Visualization technologies therefore the online resources were in abundance but not in the context of an interactive web application in relation to weather data. Regarding to the prospective of my project, It is used in the process of identifying all the individual data of a particular weather station seen at **Appendix A**, the creation of the interactive map on **Appendix B,** in the manipulation of dynamically setting the interactive year range as seen in **Appendix C** and in the creation of the different graphs through the manipulation of the data set which as mentioned before Dash is responsible for the interactivity aspect of the graphs, notice **Appendix D**.

Bootstrap 4 is used as a supplementary framework for quickly designing the front-end layout. It Is completely responsive out of the box and an additional layer of styling was used to customize some attributes, such as text and colors for the purpose of refining and highlighting specific elements.

In order to keep track of the development process GitHub as a version control platform was used, seen at **Appendix E**. This is crucial along the whole process as I was able to keep note of the changes that I have done throughout, or I could revert to older versions in case I followed a root that was leading to a deadlock. It was also helpful the task separation, for example setting technical milestones for a certain period of time regarding front-end or back-end components. That was particularly helpful to manage the time and the workload along my studies and my professional responsibilities.

3. Enhancements

There are several features that are under consideration for further improvements that would both provide more useful data presentation but also make it for accessible to the users. It is worth mentioning that based on future UX evaluation many of these elements are potentially to be reconsidered and redesigned to achieve the optimal perfomance and user's expectations. Depending on the methods that would be follow, the equivalent redesign decision will be taken for each website element. More specifically there are certain elements than need review such as the year slider.

In terms of the technical perspective there are a few improvements that have been planned but not carried not only due to time limitation but also due to the technical restrictions of the library.

The first element that needs to be improved is the main application file (app.py) that is run when the web application starts. That file contains the whole codebase of the application which as a design principle is not efficient and has resulted to rather large file with around 350 lines of code. That is because both the Front-End and call-back functions are in the same file. The Front-End code contains all the components that appear on the website, such as the map, the time slider, data table and the three graphs. The callback functions, where there are about eleven functions that are responsible for the data representation.

The solution to tackle that issue is by separating the Front-End and the Callback functions. That can be done by creating separate files that contain each one of the callback functions and then be called within the main app.py file. Unfortunately, that is not supported official out of the box by Dash therefore a couple workarounds exist from the community using separate Python functions. The challenge is that most of these solutions are very complex and would need additional installation steps that would complicate the initialization of the application as well as add extra development time which at this stage cannot be afforded.

Regarding the interactive map which contains the weather stations as embedded blue dots, which when clicked visualize the respective data. The issue is that when the user hovers over the dots the cursor doesn’t change to a point cursor, therefore it is not apparent to the user whether they are clickable. The solution perhaps is related to CSS but due to limited time it was not investigated throughout extensively. Furthermore, some key information related data could be added on during the hover event in relation to the weather station.

In terms of design aspects some further improvement could be done on the topic of information organization. For example, the heading could be redesigned by incorporating a better explanation of the web application through some basic explanatory information of its development purpose as well as more importantly guidelines for its usage. The year slider component plays a key role for the interactivity but as it currently stands there could be a few issues that do not make it very accessibly. First of all, it is not obvious what is its function as there is not any clear labelling. As it stands there is a clear technical limitation from the Dash library therefore little improvements can be suggested in its current nature.

Finally, one of the most important aspects in a website is the addition of a footer component. That would be particularly helpful in the accessibility as it would add an extra element of information area for displaying data relevant to the creation of the application, GitHub repository links as well as social media links

4. Future work

There are endless possibilities for the potential future expansion of the web application. The current state provides a solid foundation that supports development in different technical areas. The most important aspect is the possibility to add and endless pool of weather station can not only be limited to Scotland but for the whole of the UK, Europe or even the World. The problem is that it can only be achieved by the developer of the application and not by any user.

The most efficient approach to that is the addition of a custom field that allows the user to add any data set that he wants. That creates other on the other hand, such as security, data sanitation or data compatibility which is a completely different topic and need careful consideration. The positive aspect is that it is a support feature from Dash library and could be an easy addition for future development, if the relevant challenges are tackled.

In addition to the custom upload field, another interesting idea that would enhance that feature is the transformation to a complete fully custom Content Management System (CMS). That is a very interesting proposition which as far as I am concerned it has not been done before in relation to data visualization therefore an easy to use CMS platform would be created focusing around any data visualization. It could be applied in different aspects of visualization, create specific pages for each visualized area and even more create areas for discussion and exchange of ideas for the user, similar to WordPress. Since Dash is essentially based on Flask, it allows the creation of a CMS within the additional or different Flask dependences. It was slightly approached as an idea through the second coursework of Advanced Web Technologies.

Furthermore, the most important feature that has been discussed and strongly considered both by myself and by my supervisor was the possibility to compare and create correlations between not only between different weather stations but also with data fields such as rain, sunshine hours or temperature. That would particularly be useful to identify weather trends or study and compare weather phenomena. Based on the research for the specific topic, it was found out that it is very technically challenging, although it is technically supported, it would need extended development time which could not be considered at this stage.

Another interesting factor that could be a useful additional in terms of usability is the option to add the possibility of allowing the user to choose what graph method to visualize the relevant data. That creates an extra element interactive and can be specifically combined with the data comparison feature for corelating different data or weather stations. In terms of as a technical challenge it could be implemented within the current feature support and could achieved in a relevantly short period of time.

Finally, the heart of the web application is the database. The current technical implementation of the data storage is through a simple folder which contains the data sets as three separate CSV files which are then loaded to the back end. It is a very simple and straightforward approach although it creates some issues. First of all, it is not the most secure way as the files could be easily corrupted, modified or lost. Secondly it creates performance issues in case a large number of CSV files could be added in the future. Thirdly it is not certain how easy it would be to upload a new data set in the future as discussed before. Therefore, the industry standard approach is the implementation of a MySQL database that would include all the current and future user uploaded data sets. It technically possible but time consuming in the current form of the application.

5. Conclusion

This project has taught me a lot of technical skills and theory especially around Python and Data Visualization. My skills have greatly improved in Python dash and the creation of interactive web applications. My data manipulation skills were also improved through the exploration of Panda library and the understand of how CSV files work as well as through data sanitation and the identification dirty data.

It is important that through careful investigation, the careful decision-based approach of the main technology which allowed the most efficient development of the specific topic that was proposed, Weather Data Visualization. An interactive web-based application was successfully developed with a minimum set of interactive features, data visualizations and data sets which allow the users to interact, learn and identify historical data of their desired choice. The careful time management played an important role as well as the version control of the development process helped me many times speed up development.

Without doubt the possibilities of additional features are endless. With the correct skills, the available time and dedication it could potentially expand to a large-scale application which could play an important role as a data visualization platform. Alark Joshi, Researcher & Professor of Computer Science at the University of California (Joshi, 2018) , explains as a Data Scientist that there are not enough data visualization interactive applications and could be a useful tool to incorporate in their research. Enrico Bertini, Associate Professor at NYU (Bertini, 2017), argues that there should be collaborative interactive visualizations systems that would allow scientists to developer a wider perspective in data science and help analyze and societal problems.

The deliverables that were proposed on the learning contact were achieved besides the proposal of creating a mock-up design of the application, as I found that it was not needed and would be beneficial to focus on the technical aspect.

6. References

Bertini, E. (2017, November 28). *From Data Visualization to Interactive Data Analysis*. Retrieved from medium.com: https://medium.com/@FILWD/from-data-visualization-to-interactive-data-analysis-e24ae3751bf3

Joshi, A. (2018, June 13). *We need more Interactive Data Visualization tools (for the Web) in Python*. Retrieved from medium.com: https://medium.com/@alark/we-need-more-interactive-data-visualization-tools-for-the-web-in-python-ad80ec3f440e

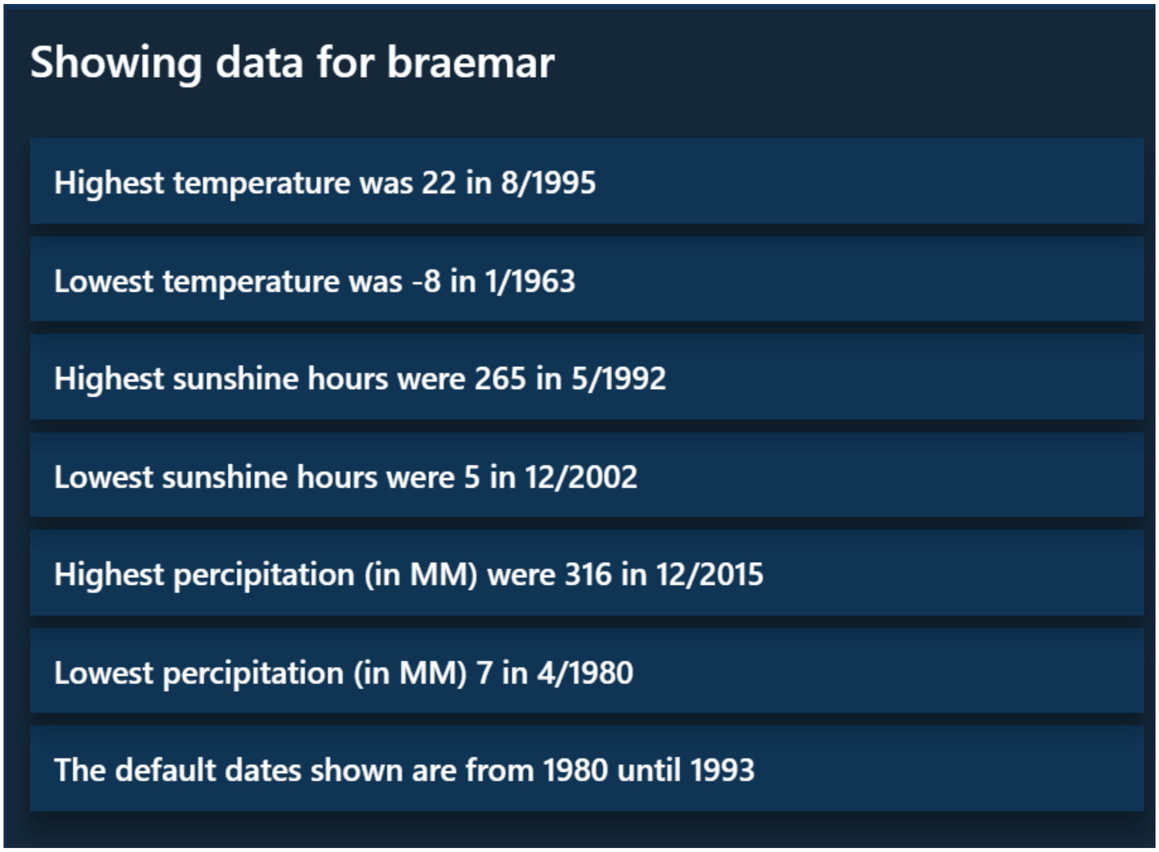
Plotly. (2018, December 9). *Build beautiful web-based interfaces in Python*. Retrieved from plot.ly: https://plot.ly/products/dash/

7. Appendices

Appendix A

List of key data presentation

This table represents that first data that appear when the user visualizes a specific weather station. That is important to identify key weather events which they could explore more in depth in the interact weather graphs through changing the year range or focusing to that particular area.



Showing data for braemar

Highest temperature was 22 in 8/1995

Lowest temperature was -8 in 1/1963

Highest sunshine hours were 265 in 5/1992

Lowest sunshine hours were 5 in 12/2002

Highest percipitation (in MM) were 316 in 12/2015

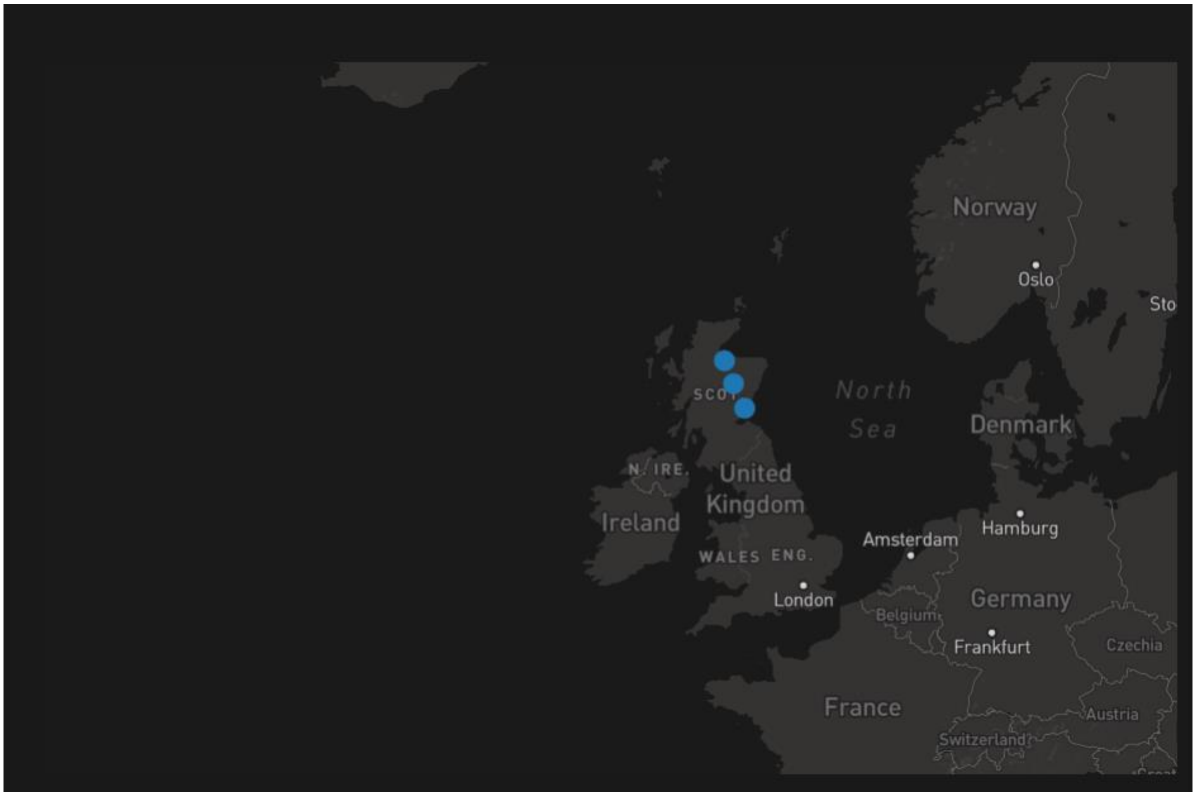
Lowest percipitation (in MM) 7 in 4/1980

The default dates shown are from 1980 until 1993

Appendix B

Interactive map

The map contains the weather stations that are available for visualization. Since it is interactive it is not static to the region of Scotland. The whole idea was for the map to be accounted for as a primary component as the first interaction approach to the users.



Appendix C

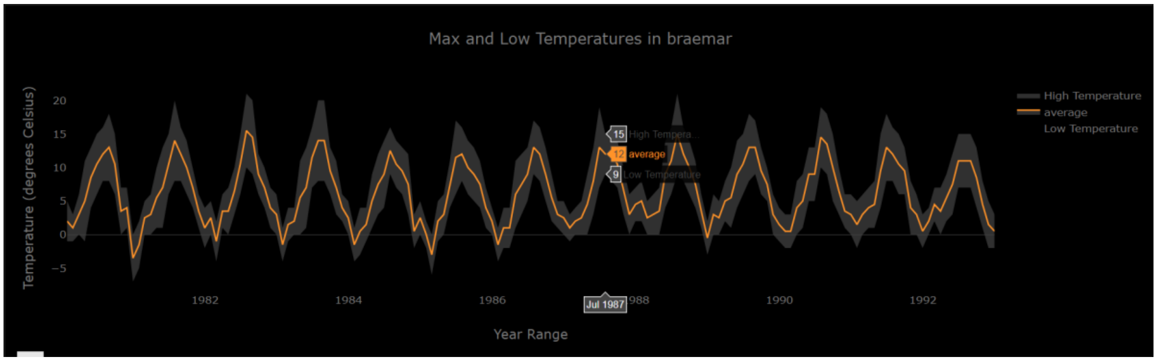
Year range slider component

This component is responsible for setting the year range that the user wishes to visualize. The user can further filer the data through interacting with the graphs, but this feature gives the user an easy approach to primarily limit the data and if desired then go more in depth in the next stage.

Appendix D

Data visualization graphs

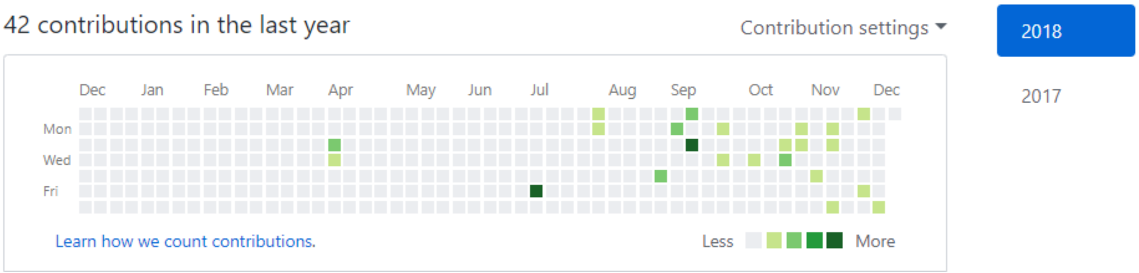
This is one of the interactive graphs that shows the max, low and average temperature, the user can easily filter, focus or hover over data to identify specific data. The graphs year range can be changed from the slider.



Appendix E

GitHub commits

This table visualizes the number of commits through the development stage, it has been in development for some time and a complete detail list of each commit can be found at the project repository which is https://github.com/angelos24/dissertation



42 contributions in the last year

Learn how we count contributions.