Data Visualisation Project Documentation

Contents

[Project Outline 2](#_Toc158724984)

[Solution Process 2](#_Toc158724985)

[Initial Visualisation: 2](#_Toc158724986)

[The 2nd Page: 5](#_Toc158724987)

[The 3rd Page: 6](#_Toc158724988)

[4th Page – AI Chatbot, ASX200 and Bitcoin Price Forecasting 7](#_Toc158724989)

[What is next for this project? 8](#_Toc158724990)

[Issues and Lessons Learnt 9](#_Toc158724991)

[Repeatability / Methodology for Solution Development 9](#_Toc158724992)

[Conclusion / Final Thoughts 10](#_Toc158724993)

# Project Outline

Data is in the background of every business, and data visualisation bridges the gap between raw data and meaningful insights. These insights can enhance understanding and facilitate decision-making in the business. This project aims to innovate our current data visualisation and analytics approach, by focusing on:

1. Combining public data with internal data.
2. Exploring new ways to visualise data such as 3D visualisations.
3. Create a benchmarking tool.

The purpose of this project is to spark transformation within HUB24 in the way data analytics is done. It will serve as an introduction to the potential value of what data visualisation tools can provide – with a plan to partner with an established analytics vendor in the future. Ultimately, we want to increase understanding across the entire business spectrum, from customers to staff alike.

Proposed outcomes:

* Enhanced visual communication.
* Informed product development.
* Partner with analytics vendor.
* Improved operational efficiency.
* Dashboard creation.

Tasks:

1. Take inspiration from industry standard products for features that could be added.
2. Source appropriate public data for visualisations.
3. Integrate data into a database format for systemic use.
4. Create visualisations from anonymised internal data.
5. Benchmarking feature to compare advisors.
6. Contain a feature that can display geographical representations.

# Solution Process

Initially, a medium for the visualisations was to be selected. The options for this were either to create a desktop application, or web application. After careful consideration of the pros and cons of both, it was decided that creating a web application would be best suited for this project – as it would allow for easier and faster development whilst keeping a stronger focus on data visualisations (opposed to technical web/app development).

Due to the many desired features to be implemented, the web app was designed to have multiple pages, each serving its own purpose to demonstrate a specific visualisation(s) / insight(s).

## Initial Visualisation:

On the first page, some basic visualisations were planned to be integrated. But to understand which graphs to use, the data needed to be analysed first. Public data was decided to be researched for as this would give more opportunity to present a diverse range of visualisations, whilst the internal data was planned to be visualised on the succeeding pages.

The internal data that was found:

1. Total personal weekly income across age groups
2. Highest and lowest average taxable income across Australia
3. Dummy data for Sydney postcodes and income

These 3 data sets, the first two are from actual sources from Australian Bureau of Statistics (ABS), and the last one is randomised data inspired from the average taxable income data.

The first two data sets were decided to be represented with bubble plots with the second one being overlayed on a geographical representation of Australia.

Firstly, the design of the first page as shown in figure 1.

Figure 1 – first page design layout:

A screen shot of a computer

Description automatically generatedThis first page consists of navigation bar at the top for directing to other pages, a GitHub link to show the readme file that gives more detailed explanation on data requirements. A drag and drop button and then radio buttons to specify which visual output that is desired. That output is presented below at the bottom of the page.

With this as the canvas, the very first visualisation can be seen below.

Figure 2 – general bubble plot:

A graph of a person with a graph

Description automatically generated with medium confidenceThe graph represents a bubble plot with the age ranges against weekly incomes. The idea behind this visualisation was to provide insight to which age groups would be best to be advertised to, when it comes to financial advising. The graph can show which group of people (indicated by the size of the bubble) is best suited to a certain product or service, based on their income and what they could afford. This is just one application / insight with this visualisation, this could also be applied to other insights like understanding the general distribution of certain groups by some parameter.

The 2nd representation is a geographical visualisation of the average taxable income across postcodes in Australia.

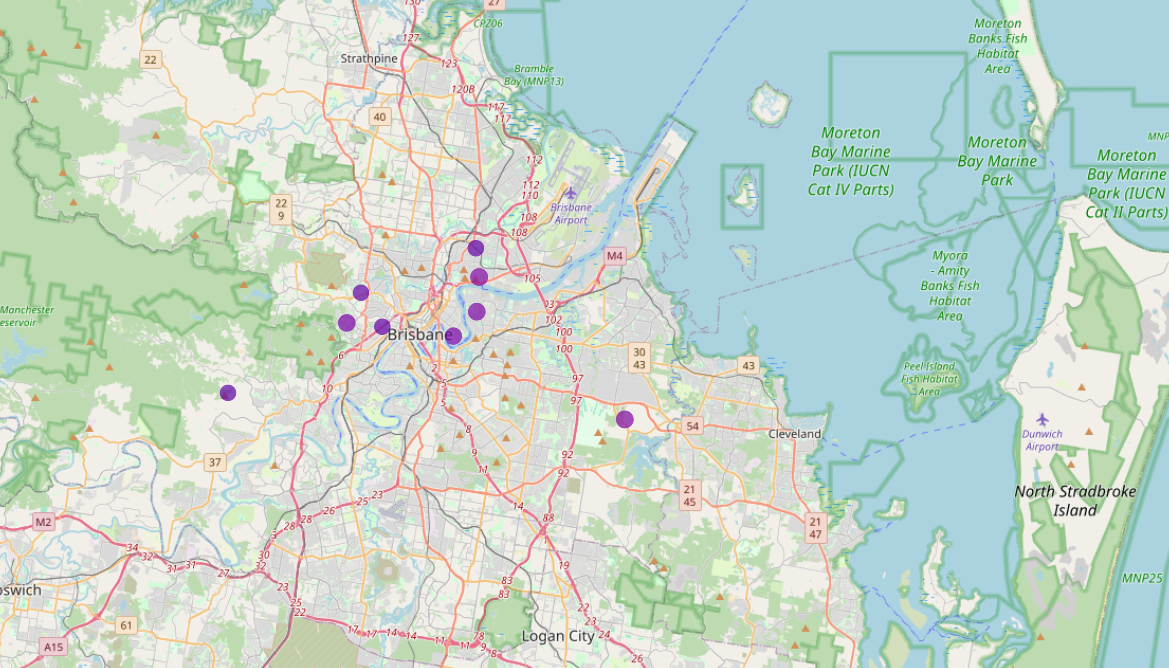
Figure 3 – bubble plot over Australia:

A map of australia with different colored spots

Description automatically generated

This graph shows bubble located in many different postcodes, most of which are not visible in this snapshot of the graph due to the zoomed-out scale of it. However, the graph shows the ranges of incomes in particular locations across Australia. This insight shows which postcodes in which states, have the most or least money, and thus which locations is best to provide financial products / services towards. This is just one insight derived from this graph, and again many more can come from these types of geographical visualisations.

Additionally, if zoomed in (look at figure 4).

Figure 4 – zoomed in geographic bubble plot:

The specific postcode locations can be seen, and the income range based on the colouration.

Now for the last visualisation for the first page, there is another geographical representation but in a more focused area (Sydney) with a hexabin plot. This graph was inspired from Door Dash’s map function on their mobile application – where show the density of which areas have a high volume of orders. In the one below, it shows the density across Sydney based on the colouration of the hexagons.

Figure 5 – hexabin plot:

A purple and blue pattern on a white surface

Description automatically generated

This data is randomised and self-generated (not real data) but is meant to show a proof of concept for this type of graph, where you can see the incomes of all locations within the rectangle.

As this was a different visual than the bubble plot, which was also able to portray the density of certain postcodes. These visualisations successfully show further insight to the given data, which can aid the process of making decisions of targeting specific areas (for marketing etc).

## The 2nd Page:

This page is the performance page, where the user can track and observe the performance of all investors within the anonymised data provided. This is one of the tasks to be completed, and it was completed through having extracted the main important information from the original CSV. Then is plotted through a line graph for each of the accounts and their corresponding closing balances. Through doing this we can see the performance of the investors and compare them to see who is performing and generating the most wealth opposed to who is performing the worst, generating the least wealth.

A graph of different colored lines

Description automatically generatedFigure 6 – line graph of all investor’s accounts:

Additionally, on the same page below where the graph is outputted, there is a leaderboard feature, where the best and worst performing account is shown.

Figure 7:

A close up of a number

Description automatically generated

This feature ultimately shows who did the best and worst, thus we use this a tool to review the how the investors are performing, which can further business operations and provide documentations for what worked and what did not.

## The 3rd Page:

The 3rd page consists of a sales funnel and asset portfolio visualisations. The sales funnel congregates the number of occurrences within the sales funnel at each stage of the sales process. But the main feature of this page is the asset portfolio shown through the spider graph. Additionally, because the spider graph only measures the frequency of the asset classes, a bar graph is shown below it which contains the actual monetary value of each asset class.

The page requests the user to upload some data on the assets that are managed, and then the user will click a button that will generate the visualisations.

Below are the sales and asset portfolio visualisations:

Figure 8:

A colorful squares with numbers

Description automatically generated with medium confidence

A blue and white diagram with text

Description automatically generated with medium confidenceFigure 9:

Figure 10:

A graph with blue squares and white text

Description automatically generated

How is the data managed?

Using SQL Lite the CSV and Excel files that are uploaded into the web application, can be converted into “.db” files. These files are then used for having a sorted medium to store and access data. Ultimately, having one medium allows for integrating many CSV files, and having easy access to the data needed for the visuals. Additionally, having the data base files allows for basic management and access to the tables of data from the original CSV’s. Ultimately, just storage and access functionalities. Obviously in more sophisticated applications, where different data tables have more complex relationships, a more proper software would be used.

## 4th Page – AI Chatbot, ASX200 and Bitcoin Price Forecasting

On the last page of the web application, it contains a chatbot sourced from OpenAI’s API key to use their ‘gpt-3.5-turbo’ model. Additionally, forecasting for the ASX200 and Bitcoin (BTC) prices are done with two different approaches. For the ASX200 forecasting a probabilistic modelling solution was done using Markov chains and the previous 60 days’ worth of data from the current date. For the BTC forecasting a more complex and effective machine learning solution was used. Namely, after careful consideration on many neural network models from convolutional neural networks (CNN’s) to recurrent neural networks (RNN’s), it was decided to use the long short-term memory (LSTM) model to predict the value of bitcoin. The choice for this model was because LSTM contains memory cells, capacity to learn long-term dependencies and very suitable for time-series data (like stock prices).

Using Google Colab, the LSTM model was able to be trained using TensorFlow python library, and an arbitrary selection of historical data from Yahoo Finance (21/09/2014 to 07/01/2016 ~ 2 years of data). The model was trained off memory cells predicting every 4th day – essentially for every 3 days of data the model predicts the 4th day, adjusting its weights using the ‘Adam’ optimisation algorithm. Then the model is fine-tuned by training it with 100 epochs (100 iterations feeding it the same data).

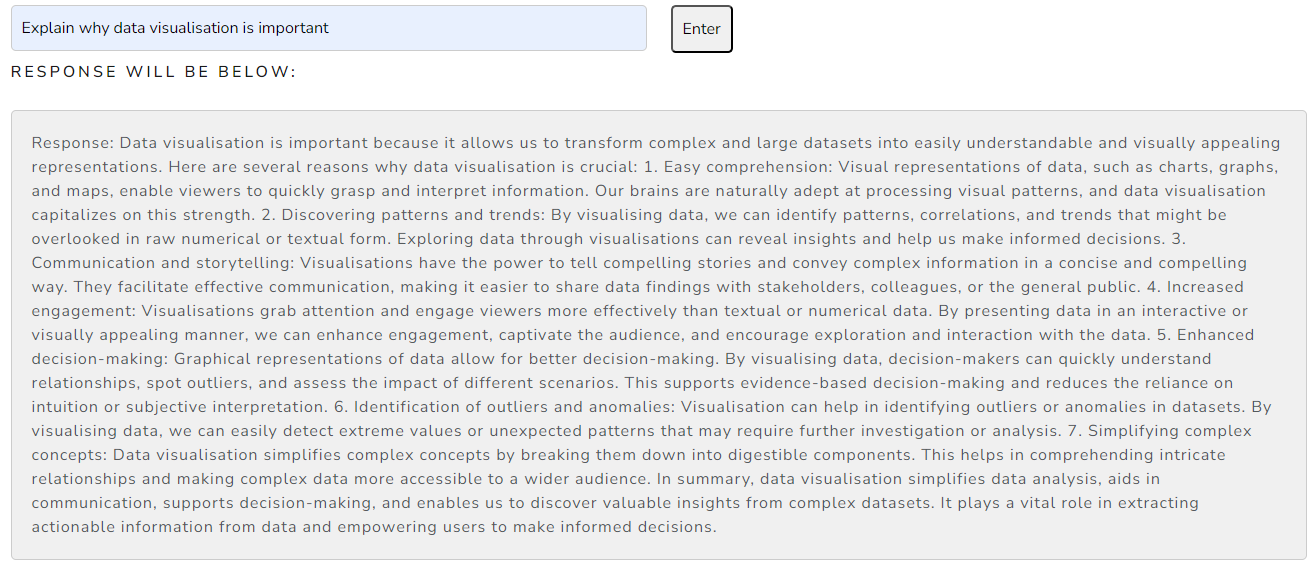
The result is shown below.

Figure 11:

A graph with numbers and lines

Description automatically generated with medium confidence

An example of the AI chatbot can be seen in figure 12.

Figure 12:

Note that the chatbot can be fine-tuned to provide responses based on a certain biasing, for example, feeding it a system bias of "You're an informative AI chatbot that specialises in data science, visualisation and analytics.". This is useful since the purpose of this chatbot is to provide an automated feature that can provide further explanations to certain visualisations and other general questions.

Now for the Markov chain solution for forecasting the ASX200 prices, an example is shown in figure 13 with it predicting 10 days into the future.

Figure 13:

A graph showing a line

Description automatically generated

It should be stated that these predictive forecasting solutions are far from accurate and are just a proof of concept to show how data visualisation and machine learning and bleed into each other. I.e. showcasing the value of visualisation when working with predictive technologies. Real-world machine learning solutions often are extremely complex and modular – based on several factors that can influence values especially within the field of quantitative trading.

# What is next for this project?

Next for this project would be to expand the depth of quality of the tools. Currently this project is mostly a demonstration of the possibilities and which type of visualisations could be best used within a financial advisory context and in other fields like machine learning. Furthering the development in this area for the business would provide even stronger insights and communication of certain data sets that have not already been taken advantage of.

This could be done either internally by having internal software developers and data scientists to produce an internal product, or partner with a certified analytics vendor. Most likely the ladder will be the best option, since this would mean guaranteed results with industry standards. Additionally, it would be more effective than hiring and developing a product from scratch. Ultimately, offloading the work to specialists but giving them specific objectives and desired visualisations that suit the business’s goals, whatever that may be.

# Issues and Lessons Learnt

Throughout the process of developing this solution, issues were experienced. One of which was the constraints from library selection. In this case, dependent on the programming language i.e. Python versus JavaScript, specific libraries are offered in the two languages. In this solution, the Plotly library provides visualisations that some JavaScript libraries do not. Additionally, the design of the visualisations is different, but can ultimately convey the same insights. So, the decision to develop this solution in the Python library was selected since it was much easier than programming in standard web development languages. This way more of an emphasis on the visualisations opposed to the ability to develop a web application. Hence, a more diverse range of possibilities could be shown whilst providing enough UI. Ultimately, the lesson learnt here is that for the true development of such a product in the future, consideration for which medium to develop in is necessary. As in the long-term time horizon of this product, consistent auditing and improvements should be able to be made – and the use of Python and the dash with Plotly library may limit this and restrict certain functionality.

Another issue encountered was the data cleansing that was required to supply the visualisations. Using the raw internal data given was not sufficient in creating desirable visualisations. The data had to be filtered for the appropriate entries etc. Thus, the lesson to be learnt here is that a level of data processing is required to operate this sort of application. This means that any vendor that is possibly partnered with in the future should be able to process data and extract the necessary entries.

Moreover, the integration of a large language models should be considered since it would be a very useful tool for chatbots to increase user experience. As well as being able to automate visualisations as an extra feature.

Lastly, the database management scheme could cause some potential issues. In this solution since it was mostly a proof of concept using SQL Lite was enough to store and manage the data. However, in larger more sophisticated applications a stronger database management software should be used. Software such as:

* MongoDB
* PostgreSQL (MyProsperity uses)
* Neo4j (graph storage)

These can do more than SQL Lite, so consideration of which one to use in a more advanced application should be made. This was a lesson learnt since whilst using SQL Lite it was notable that the ease of use was restricted since it lacked features such as stored procedures, triggers and full-text search capabilities. Also, for the full functionality of the application, ideally one singular data base is used, where the user would be able to input one large csv or multiple. This would allow for just one occurrence of inputting data. Opposed to having to upload one CSV per visualisation.

# Repeatability / Methodology for Solution Development

To repeat this solution archetype a basic procedure can be followed:

1. User Interface Design / UX:
   * Focus on designing an intuitive and user-friendly interface that enhances the overall user experience (UX).
   * Consider user research, wireframing, prototyping, and usability testing to ensure the interface meets client expectations.
2. Data Analysis / Outcome Planning:
   * Conduct thorough data analysis to understand the underlying patterns, trends, and insights within the dataset.
   * Define clear objectives and outcomes for the solution based on the insights derived from the data analysis.
   * Confirm the analysis matches with desired outcomes for the data and also where the data is going to be applied towards – like financial advisory context or machine learning context.
3. Isolate Desired Visualisations:
   * Identify the key visualisations that effectively communicate the insights and outcomes derived from the data analysis.
   * Select appropriate visualisation techniques (e.g., charts, graphs, maps) based on the nature of the data and the intended audience.
   * Ensure that the visualisations align with the objectives and outcomes defined in the previous step.
4. Integrate Visualisations into Application:
   * Develop or configure the necessary tools, libraries, or platforms to create the identified visualisations.
   * Integrate the visualisations seamlessly into the application's user interface, ensuring consistency and coherence with the overall design.
   * Implement interactive features and functionalities within the visualisations to enhance user engagement and interactivity.
5. Deploy with Cloud Service (e.g., AWS) or Desktop Application:
   * Choose an appropriate deployment option based on factors such as scalability, accessibility, and security requirements.
   * Deploy the solution using cloud services like Amazon Web Services (AWS), Microsoft Azure, or Google Cloud Platform (GCP) for web-based applications.
   * Alternatively, consider deploying the solution as a desktop application for specific use cases or environments.
   * Ensure proper configuration, testing, and monitoring of the deployed solution to maintain performance, reliability and satisfying client needs.

This step-by-step process provides a structured methodology for developing and deploying solutions based on data analysis and visualisation. Following this methodology can lead to the creation of effective and impactful solutions that deliver desirable outcomes for users and stakeholders.

# Conclusion / Final Thoughts

Expanding the depth and quality of the tools used in this project represents the next step in leveraging data visualisation to drive insights and decision-making within the financial advisory context. By partnering with a certified analytics vendor, the project can access industry-standard products and expertise, ultimately enhancing the value proposition for the business.

Moving forward, it will be crucial to continue refining and iterating upon the visualisations and analytical techniques employed in this project. This includes exploring advanced visualization techniques, incorporating machine learning algorithms for predictive analysis, and integrating real-time data sources to ensure the relevance and accuracy of insights.

Furthermore, ongoing collaboration with stakeholders and end-users will be essential to validate the effectiveness and usability of the visualisations, ensuring they meet the evolving needs and preferences of the target audience.

In conclusion, by embracing data visualisation as a strategic tool for informing and guiding decision-making processes, the project lays the foundation for more informed, data-driven decision-making within the financial advisory domain. Through continuous improvement and innovation, the project aims to unlock new opportunities for enhancing operational efficiency, mitigating risks, and delivering value-added services to clients.