NOVA IMS
Information Management School

MAA

Mestrado em Métodos Analíticos Avançados
Master Program in Advanced Analytics

Data Visualization

Final Project - World Tour Simulator

Alex Anthony Panchot (M20190546)
Bruno de Lima Vieira (M20190922)
Hugo Saisse Mentzingen da Silva (M20190215)
Leonardo Motta Perazzo Lannes (M20180036)

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

Dataset Description

We are using several different datasets for this project. The main file is "worldcities.csv" which lists the city's name, state, country, longitude and latitude, and population. As we wanted supplemental data for each city, we found and added several other datasets. Another dataset is "wiki_international_visitors.csv", which ranks the top 100 cities based on their number of international arrivals. As this is a travel-oriented application, we decided that looking into the number of arrivals could be important as more arrivals implies more tourists, for better or worse, depending on the end user's preferences.

Another relevant city indicator that is important for tourists is the cost of a single-way ticket on public transportation. In the "transportation_costs.xlsx" dataset each city's single-way ticket prices are listed in US dollars. Like transportation fares, hotel costs are important for travelers so these are listed in the "average_hotel_prices.xlsx" dataset.

Since some cities are quite close together, having both cities on the map causes their markers to be overlapping with markers of other cities. In order to avoid this, we only allowed cities with enough spacing to be available for selection to the user.

Considering that the transportation and hotel datasets did not include data for every city (like Lisbon), we manually found and filled in the relevant data. Finally, all of the data that we used is considered "Sequential" as there cannot be negative values.

Visualization and Interaction Choices

The inspiration for the work came from using travel websites as well as our previous experience working with genetic algorithms (GA). While travel websites usually do not list information for multiple trips (usually such a trip is impractical and too expensive), we thought it would nonetheless be interesting to see how such a journey would look like. Also, if you were to fly to many cities across the world, you may want to see the shortest path available. This allows you to fly for fewer hours and spend more of them sightseeing.

The shortest path is not so trivial to see for many cities and that is why we are implementing a GA to solve this problem (this problem is akin to the "traveling salesman problem"). The technical implementation for this is explained in the "Technical Aspects" section. When the GA finishes, it creates an animation of the evolution of the shortest path from a random initialization to the optimal solution. The lines that connect the cities in the map change their width according to the 'fitness' of each algorithm iteration, which means that the line becomes thinner as the total traveling distance decreases.

As this is a travel website prototype, the main focus for users is to select different combinations of cities and to see the results for the different indicators and graphs. The users are presented with several different interactivity options including dropdown menus, slider, tabs, hovers and selectable captions.

The color choices were chosen with a dark theme in mind. This allows the user's eyes to be quickly drawn to the visualizations as white on black stands out better than the inverse. By using other primary colors such as red and yellow the text stands out from the white selection boxes. Because of the graphs that we used, we could not use more "effective" channels than size and color to express each attribute. Despite this, the attributes are still clearly defined.

The graph scales and spacing were set to occupy all the space available in each screen, taking into consideration that Dash resizes the elements according to the user's browser. Tabs were also used to avoid scrolling the page and to retain the focus of the customer.

Reading the Visualization

For most of the information, the data is numerical and is graphed as such. However, we do use a few data encodings. The city rank is derived from the number of international arrivals, with the smallest rank representing the largest number of arrivals. For the markers (each city) on the map, their color is a visual representation of their rank. The markers for the scatter plot are created with the color representing the city's continent and their size representing their rank (the bigger the better). The opposite is not possible as the continents are categorical and should not be represented by differently sized markers. The encoding used in the GA is described in the Technical Aspects section.

As this is a travel application, we want the user to be able to decide which cities they want to look at. Initially, the app has a few preselected cities, but the user is able to delete and add different ones. As we do not have data for every possible city, the user can see the available cities from a dropdown menu. As the user changes their city selection, several indicators change in real time. When the user is satisfied with their selection of cities, they can press the "submit" button to pass the list of cities to the GA.

The algorithm runs in the background and when finished, it makes the map's animation available. When the "play" button is pressed, the different possible paths are shown until the optimal path (least distance) is found. We believe that animating the map with the algorithm search can help retaining the customer attention.

There is also a dropdown menu where the user can select one of several indicators. These indicators allow for a ranking comparison between the selected cities, which is displayed in a bar chart.

A slider was also added, wherewith the user can select a cut-off for the rank of the city. As mentioned before, this rank is ordered by the number of international arrivals, so that the user can choose to travel only between the most popular cities. As the user moves the slider back and forth, the cities that meet their selected criteria are displayed in a box to the right of the selection pane and submitting the route calculation will obey this choice.

In addition to the map and real-time indicators, a bar chart displays the selected cities and the selected indicator. Hovering over each bar gives an exact value for the indicator. Below the bar chart is a bubble scatter plot of the hotel vs transportation costs. The size of the bubbles represent the rank (the best the rank, the bigger the bubble) and the color of each bubble represent the city's continent. This visualization can help the customer on making continent or rank-based choices.

Storytelling

The report follows a "martini glass" style as it leads the user directly into the user changeable settings, with these placed at the top left. Directly to their right, the user can see the printout of their selection that changes immediately. Below these two sections are tabs where the user can find more information after they confirm their selection. This layout also follows the left to right, top to bottom pattern that English is read in, matching the human focus behavior.

Technical Aspects

The project code was initially sourced from the code used in the practical class as it contained working dash, html, and CSS code. From this working template, we rearranged the containers and added the different graphs and map from the Plotly package.

The code for the GA begins by taking a distance matrix of the different cities and encoding each city as a sequential number. The value of each city is not important as the value is mapped back to the city's name at the end. Different combinations of cities are then randomly created and run through the algorithm.

The algorithm works by mutating and crossing over different combinations of cities and only keeping the best solutions (lowest distance in this case) for the next generation. It then repeats this cycle until it reaches 1000 generations. The generations, as long as the solutions, are saved for further use in the map animation. The GA code is in a separate file (ga.py) from the main application.

The code for the project is on GitHub

Discussion

In this application, a lot of what is accomplished is “under the hood” so to say. The GA itself does not appear graphically, but it does do a lot for the user in that it is able to take a list of destinations and within a few seconds inform the user on the path with the shortest distance. Beyond just the best path to take, the user has various indicators and pricing data available to them. This helps organize a traveler’s budget.

The limitations of the application come from the unfamiliarity with CSS and HTML code as well as the Dash platform. CSS and HTML knowledge helps with the colors/styles and placement of items within the application, thus lack of this knowledge can make simple things such as changing the color or size of a specific box challenging. The interactivity with Dash and the callbacks are still challenging as numeric computational coding does not usually require inputs while the code is being run.

A limitation of the arrivals dataset is that it counts any person who arrives in a city and stays at least 24 hours. As this count everyone, the data can include workers who come to a country to work and are tourists. For example, Hong Kong is at the top of this list and would be an example of a country where foreign workers cross the border to work.

We can also mention that obtaining attributes to enrich this work was challenging as we observed a lack of structured data. Average flight costs between cities, for instance, seemed impossible to find.

For the future, more work could be put in making fancier coloring such as gradients or shadowing. The map animation could also be improved with perhaps a small plane that follows along the path. Also the map could be made such that it is interactive and clicking on a city would select it. With that, improvements to the GA must also be made so that the computation time is reduced. This allows the user to try out different combinations and to see the results in real-time.

More data can also be sourced such as real-time pricing and time data for flights. This could allow for the algorithm to find the cheapest or shortest time flight path rather than just distance.

References

Simplemaps.com. (2020). World Cities Database | Simplemaps.com. [online] Available at: https://simplemaps.com/data/world-cities [Accessed 15 Jan. 2020].

Yasmeen, R. (2020). Top 100 City Destinations 2019 Edition. [online] Euromonitor International. Available at: http://bit.do/fpgXM [Accessed 15 Jan. 2020].

Taxi2airport.com. (2020). Research 2019: Cost of public transportation in 53 countries. [online] Available at: https://www.taxi2airport.com/en/blog/cost-public-transportation [Accessed 15 Jan. 2020].

Trivago Business Blog. (2020). The trivago Hotel Price Index - Track Global Hotel Pricing Trends. [online] Available at: https://businessblog.trivago.com/trivago-hotel-price-index/ [Accessed 15 Jan. 2020].