1.0 Business Problem and Target Market

The first section of the document discusses the nature of the business problem and identifies any stakeholders that may be interested.

1.1 Background

Deleted:

New York city is a global destination, cosmopolitan "wonderland", world icon and, as of late, hosts more than 60 million visitors a year. This is, probably, exclusive of those on "short" business trips. It is, often, difficult for travellers to determine where to stay if they are free to choose. The sheer number of "attractions" may pose logistical nightmares. Given travellers have different tastes and preferences "generic: guides (such as books or the Internet) may be overwhelming (and at the same time, "insufficient") in suggesting accommodation to try and make the most of, sometimes "limited", time. It is, therefore, advantageous to have something that will recommend venues that attempts to minimise physical distance to expressed "Points of Interest" (POI).

1.2 Problem

It may prove difficult for travellers to determine where to stay given the "ironic" psychological concept of the "tyranny of too much choice". Visitors may want to "maximise" their time by selecting locations that are geographically proximate to venues they would like to visit. The primary aim of this project is to help the user plan to stay where the total physical distance of all POIs is considered. This is specifically broken down in the next section.

1.2.1 Aims, Objectives, and Benefits:

1. Provide the neighborhoods and boroughs of hotels (or their equivalents) with the least total geographic distances. This may help reduce costs by making POIs either walkable or "close by" if transportation need to be used.,

2. Identify the number of nearby venues. This may make a location more "desirable".

Aside from travellers, this may be of some interest to the New York city tourism board

Commented [RA1]: You didn't really state the benefits…what makes this different from existing websites that compare accommodations?

> (https://www.viator.com/New-York-City-tourism/New-York-TopToursandActivities/d687t30053?semLander=true&m=28353&supag=6517262291&supsc=kwd355514978618&supai=273084059420&supap=1t3&supdv=c&supnt=nt:g|clk:EAIaIQobChMIlKKwzfWS5AIVmg4rCh0nnAnvEAAYAyAAEgLcvD_BwE&suplp=9070825&supli=&supti=kwd355514978618&tsem=true&supci=kwd355514978618&supap1=&supap2=&gclid=EAIaIQobChMIlKKwzfWS5AIVmg4rCh0nnAnvEAAYAyAAEgLcvD_BwE).

There may be other benefits that accrue once the final deliverable is submitted but it is essential that certain project objectives be made explicit so it would be possible to gauge the efficacy of the outcome. I have observed a tendency to overpromise and underdeliver to boost the chances of securing a contract. Since the financial pressure does not technically exist, it seems more prudent to offer a realistic proposal rather than have an overoptimistic schedule or an impractical set of deliverables.

Commented [RA2]: Is this statement really needed?

1.3 Other Pertinent Factors to Consider

The following talks about how other aspects can affect the planned work.

1.3.1 Justification

Since New York city was used in a previous lab, the investigator can use (or reuse) various informational resources and code to try to build on (and leverage) previous assets. Moreover, my family and I visited New York and it would have been "good" to have had a resource such as this that would have been instrumental in our planning process.

1.3.2 Assumptions

A "heavy cluster" of hotels in an area can be considered a "good" location as this is can be taken as a proxy for proximity to "popular" sights or amenities (as chains often survey the area prior to commencing building). Moreover, an "ideal" place to stay is mainly determined by physical distance.

1.3.3 Constraints

Different categories for accommodation were purely dictated by the classifications in the FourSquare database. Under Hotel are Bed & Breakfast, Boarding House, Hostel, Hotel Pool, Inn, Motel, Resort, and Vacation Rental.

Being a person with a disability, I wanted to include accessibility. However, I noticed the differences in nomenclature in my travels to America and Japan. Despite also being developed economies, the facilities were not as accommodating as Australia and the terminology used often exhibited differences. While a "good" idea, it was impractical and difficult to incorporate this aspect.

While the sharing economy apps (like AirBnB) are not currently supported (as far as I'm aware), future iterations may also consider these non-traditional forms of dwelling.

The outcomes for the overall set of requirements also need to be considered so the alignment of outputs can be consciously taken into account. This not only serves to guide the validity of the development but can act as a potential roadmap to possible future directions or entirely separate branches.

Predefined (later on allowing users to specify their own interests) POIs will allow for the amalgamation of distances to various lodgings (i.e. the total distance from the shelter to place they want to visit). While not a full-blown recommender system, it can suggest various accommodations (factoring in the "nearest" Boroughs and Neighborhoods of New York city) given their proximity to certain venues. Regardless of whether they are potential tourists or are on a short stay business trips, it endeavours to provide users with places to stay to try to get the most out of their visit.

1.3.4 Software Development Issues

A Jupyter notebook stored in a Skills Network lab was purposely chosen. This was explicitly done for several reasons: 1.) This was the format used in the course and familiar to peer reviewers. Moreover, it has the benefit of being "submit ready" and could easily be exported to a text format., 2.) This allowed for the flexibility to use Python code to test concepts (as it is more important to determine feasibility and specify requirements)., and 3.) IBM's backup regime is probably more comprehensive than what I personally have set-up.

Unfortunately, my experience was less than stellar. There were four major instances of failure: when I returned to my Jupyter notebooks I noticed “chunks” of code missing or previous data reinstated. To get around this, I downloaded a copy of my notebooks and exported PDF versions at major junctures, but it was too late for certain portions which I ended up recoding and debugging – wasting valuable analysis time. Moreover the kernel would “randomly” restart at times, causing me to ”lose” things I had worked hard on – I’m not usually one to complain but the sheer effort to produce a line that invariably goes “missing” can be very frustrating. In short, the quality of the tool can vastly affect the outputs generated.

Despite the requirements being mainly text-based (and formats like MS word, .txt, .rtf, or .pdf would have sufficed), further validation and the importance of feasibility became more crucial given the "shorter" time frame. Python code was generated for analyses and not simply producing a “product”. Based on my prior, personal experience, 3 months is often a reasonable period for software development. Given I had only 2 weeks to complete the capstone project (aside from my other commitments) a Design-to-Schedule methodology (that is, given a fixed deadline do your best to deliver what is possible) will be used. The result is a "Proof of Concept" prototype - typically the practice is be prepared to throw the first one away.

I found the allocated time of two weeks for the assignment rather tight. Having not been a developer for over a decade now I found the work extremely challenging. Most of the errors I encountered were syntactical or quirks of the language rather than logical. Not using Python regularly and my exposure to other programming languages at times served as a hindrance. Two weeks to do either analysis or programming was more reasonable for me given my other obligations: certainly not as a joint task including the plethora of output requirements. When practicable, external sources were tapped but sadly there was not enough time for this “luxury”.

2.0 Data

The next portion talks about what data was used in the context of the problem identified.

2.1 Data Sources

A CSV file containing New York data (i.e. Borough, Neighborhood, Latitude, and Longitude) was loaded into a Pandas DataFrame to act as the primary source of data and help facilitate any further processing required. Moreover, a POI csv file was manually constructed so it can read into a Dataframe. Furthermore, the FourSquare API was used to get the hotels and other nearby venues which helped analyses of the retrieved data and provided the ability to "drill-down" to help aid the quality of, or yield further, insights.

A few dataframes were also saved as CSV files. This made a lot of sense since it helped minimize calls to the FourSquare API and saved a lot of time rather than always having to invoke Python code to build the data. As a case in point given both considerations, data about New York hotels were stored.

External websites like LatLong.net (https://www.latlong.net/convert-address-to-lat-long.html) to get the Latitude and Longitude of POIs. And Movable Type Scripts (https://www.movable-type.co.uk/scripts/latlong.html) to help verify that the calculations using Python to compute distances given a start and end set of coordinates (in Latitude and Longitude) and correctly implemented the Haversine formula.

A Wikipedia page
(https://en.wikipedia.org/wiki/Tourism_in_New_York_City#Most_visited_attractions)

was also used as the basis for the POIs. As most had collected cumulative visitor information, these were used over yearly data which may fluctuate for various reasons. The top five were used to help reduce the computational load, as well as the cognitive load given the timeframe.

The top 5 are as follows (and their to-date number of visitors are indicated):

1. Central Park 42m

2. Time Square 39m

3. Grand Central Terminal 21.6m

4. Theater District (including Broadway) 13m

5. Rockefeller Centre (including 30 Rock)12.8m

The 9/11 memorial is not in the top 5 given its recency. Moreover, it makes sense that the Statue of Liberty is well above the top 10 as only ~20% of the 65 million visitors last year were international.

Given the number of columns returned by FourSquare, I resorted to reducing the number of POIs further than first thought (and tried dropping more hotel columns) as it was taking some time to process.

2.2 Data Cleaning

To be safe, all DataFrames were checked for redundancies and any duplicates found were eliminated. Seeing as the data was relatively “clean” (which was a huge factor for it being chosen), minimal, or hardly any, effort for data cleaning was necessary (unlike what was done in the previous assignment).

When I visually inspected the hotel data from returned from the FourSquare API, I noticed a few ‘misplaced’ entries. I manually eliminated venues that were part of the hotel or one that seemed meant for canines.

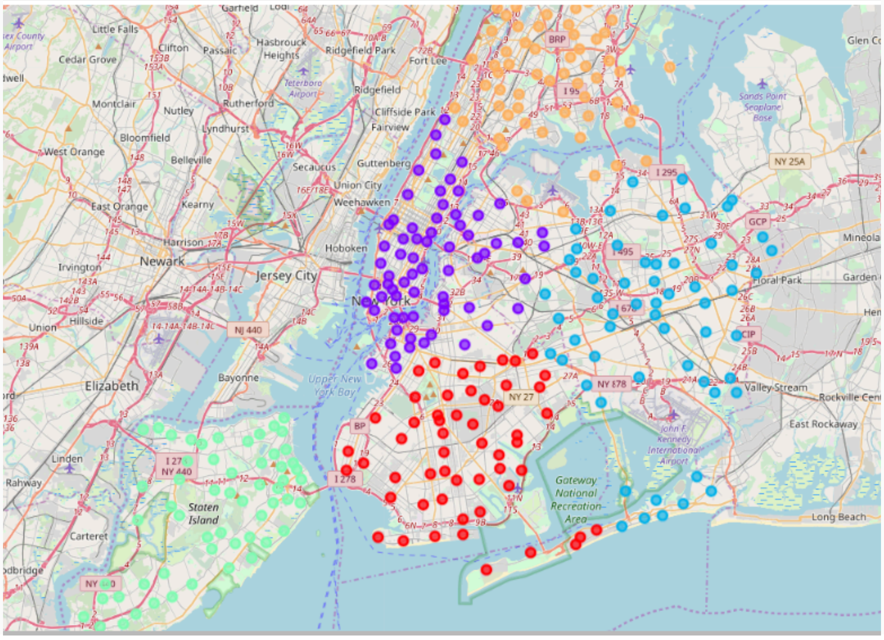
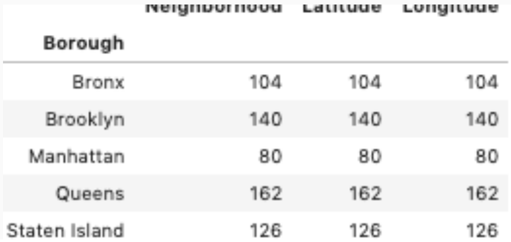


Figure 1 - Boroughs vs. "Clusters"

That said, columns with categorical data can be transformed using one-hot encoding to enable the use of Machine Learning functions built into Python (since I'm not a statistician). Particular attention was paid to the Neighborhood column because of the sheer volume of different values. Ultimately, it did not make sense for me to use one-hot encoding as the clusters used seemed to match the Boroughs of New York city based on a visual inspection of the generated Folium map (Fig. 1) and counts provided by the groupby (Tab. 1) function of the dataframe.



	Neighborhood	Latitude	Longitude
Borough			
Bronx	104	104	104
Brooklyn	140	140	140
Manhattan	80	80	80
Queens	162	162	162
Staten Island	126	126	126

Table 1 - Neighborhoods grouped to Boroughs

2.3 Feature Selection

Folium maps (or other “meaningful” Data Visualization methods) was used to represent the contents of the New York city DataFrames and help evaluate candidate accommodations (and their respective Boroughs and Neighbohoods). Moreover, they were used to help clarify if the clusters "roughly" conformed to the 5 major boroughs of New York city (namely Manhattan, Brooklyn, Queens, the Bronx, and Staten Island). This helped determine the relevant columns, refined the area to search, and reduced the number of rows of the dataset (when possible).

The final number of columns comprised the feature set and helped determine the dependent, as well as, independent variable(s). Selection was influenced by the contents of the FourSquare database. Of interest are which columns correlate with the Haversine formula so that a predictive model can be tried to be built.

These were mainly based on the hotel columns returned during a FourSquare API search. What seemed like “irrelevant” (e.g. referralId, hasPerk, etc.) or redundant (e.g. categories, location.labeledLatLngs, etc.) columns were dropped. That said, these columns could have been investigated given more time given there were possible “hidden” correlations.

2.4 Other Tools

Although a best effort was made to appropriately incorporate the Data Science technologies covered in the course videos (or used in the labs), if time permits other previous courses will be investigated, if any other tools can help enhance the functionality or analyses that were inadvertently omitted as changing requirements may arise that are not always possible to consider in advance. Particularly, the other ML techniques may be useful.

Time ran out for further data analyses despite initial findings.

3.0 Methodology

This part discusses the “how” rather than the “what”.

3.1 Modelling and k-means clustering

Modelling was primarily through segmentation (i.e. classification). K-means was used to form the clusters.

5 clusters were chosen as the number matched to the 5 main boroughs of New York city. This was a “fortunate” choice as there turned out to be a direct correspondence between the two.

Had there been more time, it would have been interesting to see how a mathematical model (like regression) would perform as compared to what was used. Sometimes it is just a preference or dictated by the tools available, but one can be more suited given a particular circumstance. Clearly, it would have been more ideal to have conducted both. Exploration of and the choice of particular feature sets would be more vital in the latter case.

Initially, a separate cluster was supposed to be performed on the hotel but doing so became less significant after plotting these on a map.

3.2 Additional Columns

Originally, several “computed” columns were envisaged that may have yielded additional insights. Firstly, a count of all the POIs. This not only “generalized” it but afforded some variety as the POIs were system, and not, user dictated for the sake of simplicity. Also, a column noting the maximum physical distance from the POIs to the hotel might have been useful. Moreover, a count of the number of other hotels “close“ by was planned. This led to a question for the standard that constituted “nearness” but in the end was rendered moot by what appeared to be the “visual” proximity of most hotels (as they seemed to be in a tight “cluster”.

In order to meet the outlined objectives only two columns were added. The first to store the total Haversine value. And the other, to store the count of the venues nearby. I ended up only adding these two columns due to time pressure. While I met my objectives, my output was below my usual standard. While there were slight variances to my Haversine figures, all were at about 44 km, making the variations unremarkable. As for the nearby venue count, I got 30 for all. I even did these several times and manually did the last one to check. This made sense since they were "clustered" and I suspect the differences lie in the distances of the venues. Seeing, essentially, as all hotels meet the criteria, it may be worth looking at the ratings (and the reputation of those users) to further differentiate these.

4.0 Results

The next part outlines the resulting output.

4.1 Latitude & Longitude Differences

There was, I thought, a negligible difference between the Wikipedia (https://en.wikipedia.org/wiki/Tourism_in_New_York_City#Most_visited_attractions) and the LatLong (https://www.latlong.net/convert-address-to-lat-long.html
) pages – a 3rd or 4th decimal discrepancy in coordinates. Understandably, there were only obvious discrepancies used (I suppose differences in interpretations or opinions came into play) instead of a specific location/area – I'm more inclined to use a representative address. Moreover for Parks, there can be differences to the 2nd decimal place but I suspect that is due to disagreements regarding the center of the park.

I chose to use the latter website because:

1. Data from Wikipedia can sometimes be unreliable as "unauthorized" individuals are allowed to make edits,
2. The latter site was previously identified as a data source, and
3. It uses (and displays) Google Maps which can be comparable to Foursquare.

The results did not seem significantly affected by this choice.

4.2 One-hot encoding

This technique seemed irrelevant for use in clustering of neighborhoods as the segments seemed to correspond to the distribution of the 5 Boroughs of New York city.

Since no clustering was performed for the hotels either, it seemed wasteful to proceed with this. That said, initial groundwork was prepared if any of these became necessary.

4.3 Merging

At first, I merged the "skinny" hotels (by significantly removing columns) with a truth table of 5 variables corresponding to the top POIs. Though not ideal, I reduced it to 5 hotels with 8 columns and 3 POIs. I reduced the numbers further, but this was still problematic so considered other alternatives given the fast approaching deadline. After trying desperately to figure out how to balance the loop with a "reduced" dataset, I decided to go a different route.

4.4 Radius

Since the FourSquare API often asks for a radius, a value of 4024 was used. Since this is in meters and the lower bound for what a typical person walks each day, assuming 8 hours, this is the average distance a person can cover in an hour (assuming catching some sort of transportation is a possible alternative). This seemed to return reasonable results and is not “far” from the 5k used as a parameter in labs.

5.0 Discussion

Despite all my caveats and software experience, I was still overidealistic in what I wanted to do. This discusses some aspects of the results and where to go from here.

5.1 Findings

All the hotels retrieved had Haversine values of 44 km from all POIs so any hotel is acceptable via distance.   That said it would have been interesting to note POI with the greatest Haversine distance. I was able to verify this fact as I changed my approach during the last day before the deadline - partly due to necessity and partly because of what I saw during the interim. I opted to abandon my original plan to “merge” a dataframe of a “truth table” that would correspond to the top 5 POIs: given 28 hotels and a number of “undropped” columns, this equated to just over 8 thousand iterations just to construct an amalgam dataframe that proved too computationally “expensive” (as it took a while for it to generate a result which was not necessarily correct). Instead I still opted to include all 28 hotels by adding a computed column to it and computing for the total Haversine distance separately.

The similar results were not very surprising given the close “clustering” of hotel as shown visually below (Fig. 2).

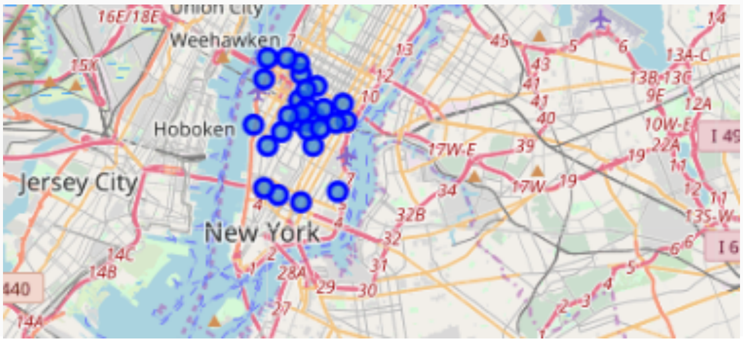


Figure 2 - Hotel Locations

The nearby venues were also ‘tied’ at 30 (I surmise from their “close” clustering. Frankly, I was eager for more variety here as I was hoping to be able to sort these and come up with some kind of ranking system. Their distances to stated, preferred venues seem a more logical choice in hindsight.

In essence, this confirmed the “implicit” knowledge of locating the hotel “near” popular attractions made good business sense. Subjecting the data to “clustering”, I suspect would offer minimal, if any, insight in to the relationship between a particular hotel and various POI. I suppose if you wanted to be pedantic about it you could sort them from least to greatest physical distance but most differences would only be a few meters.

In general, the hotels were in Manhattan and, not surprisingly, fared “well”. This made sense since 9 out of the top 10 POIs were located in this Borough. This, probably, explains a lot for the chosen locations of the hotels.

5.2 Recommendations

The work can be improved. Like anything else that has humble beginnings, with some work it may end up as a useful tool.

5.2.1 Other Analyses

Some mathematical modelling can be done, and some other Machine Learning techniques can be employed so there are more “robust” results.

Had there been more time, a determination of a better form of analysis might have been possible. Because the Haversine formula is an objective measure, it is “simpler” to test for the efficacy of predictive models. Unfortunately, not all problems are as straightforward as this.

5.2.2 (Web) Scraping

In the future, maybe some kind of scraping can be used to help ensure that some of the source data is always up-to-date.

5.2.3 User Input and POI

Since people have different priorities, it would be more relevant if the user would be allowed to specify which (and how many) POIs rather than just having a fixed list. Perhaps “weighting” or ranking POIs can make for a more “comprehensive” model.

6.0 Conclusion

Although not what was originally conceived, it has the potential to be a pragmatic solution. This confirmed empirically what we suspected and simply took on in “faith”.

6.1 Summary

This proved more an exercise of analysis rather generation. The Python code ended up furthering analyses considerably and not for developing a “product” as first conceptualized. The results may be “obvious” but they were confirmed – at least they were consistent with expectations.

Sure, I sometimes (inadvertently) “doubled-up” in certain parts of this document or “flow” and what seemed natural to me but for the most part adhered to the requirement of the sections dictated.

The stated objectives were met but a sense of wanting it to be more persists. Sure, the course (and most subjects in the certificate) can be better structured but I did learn the “nuts and bolts” of data science. Unfortunately despite my best efforts, this document is not purely about the capstone project.

6.2 Future Directions

Like everything else, it would greatly benefit by making it more mobile. This would not only increase the customer base but may be easier to use as the affordances and the usability requirements of the design dictated by mobile technologies may have some impact.

It can be further improved by crowdsourcing the main code and any suitable enhancements made if the goals are similarly aligned. In cases where the intent is different or there is an alternate vision for it, it may yield other branches that were not originally even considered by the proponent.

Deleted: (old spec) ## 2.) A full report consisting of all of the following components