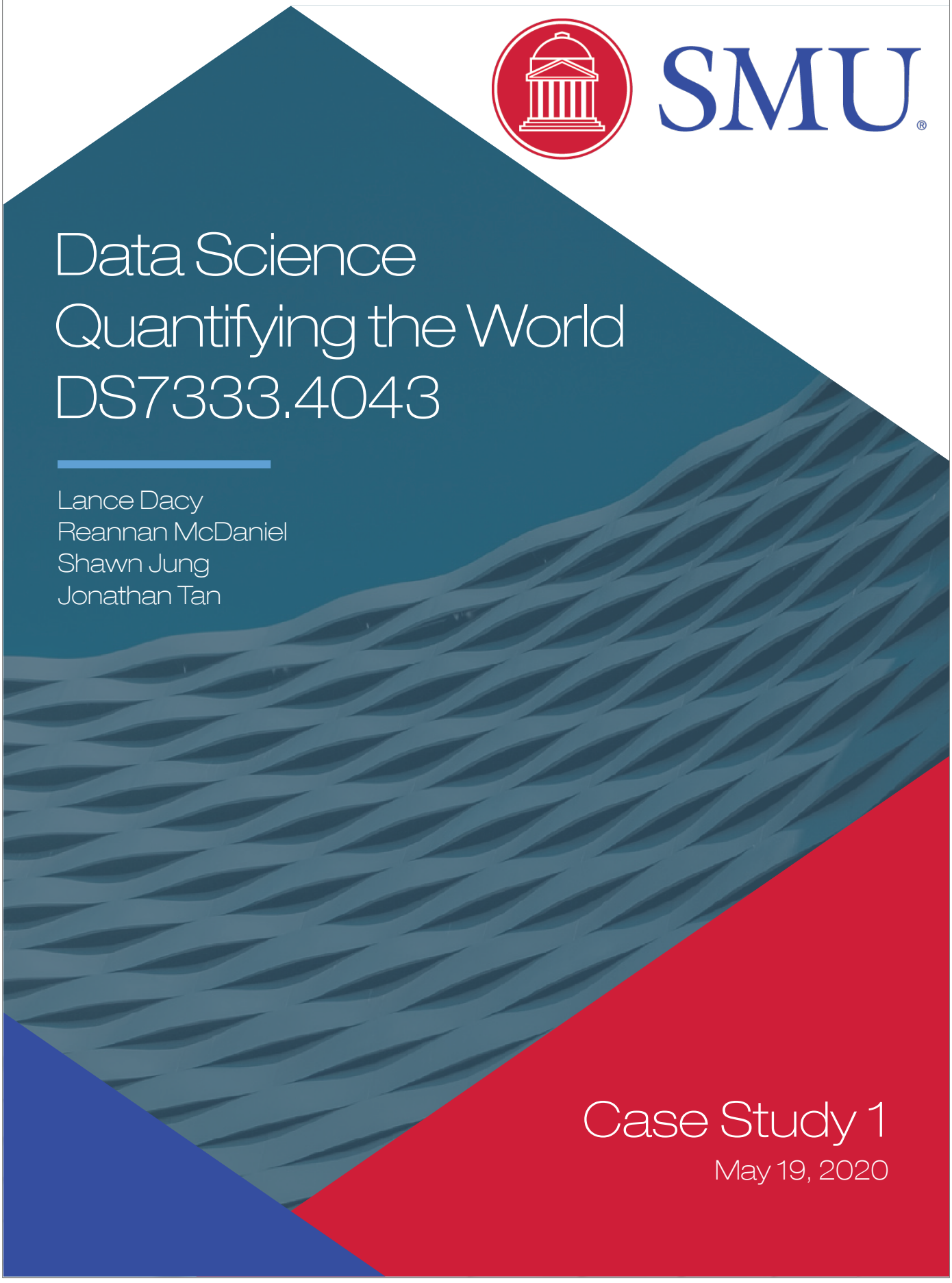


SMU

Data Science Quantifying the World DS7333.4043

Lance Dacy
Reannan McDaniel
Shawn Jung
Jonathan Tan

Case Study 1

May 19, 2020

Table of Contents

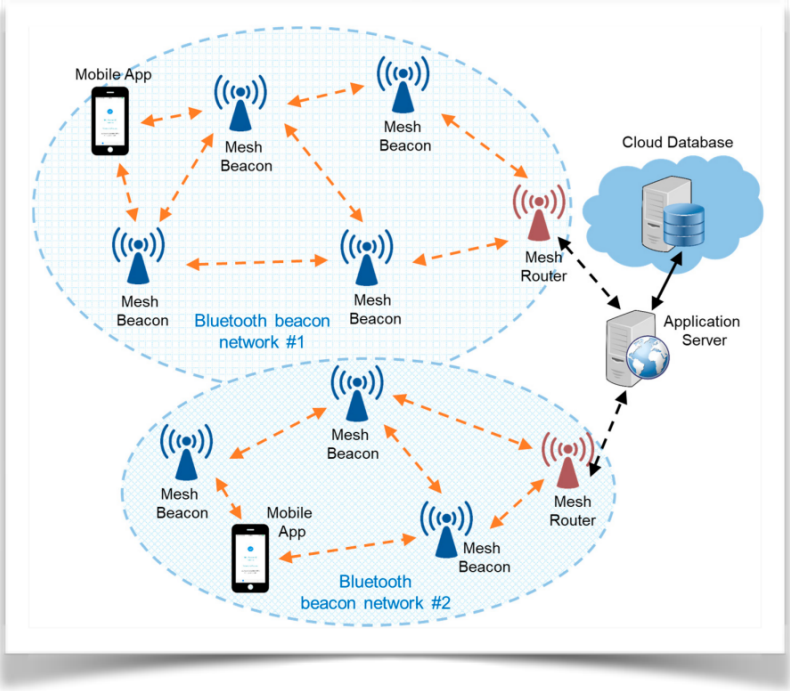
Introduction

The wireless networking industry has generated commercial and research interests in methods that use unique protocols to serve as basis for data capture. While GPS has been around a long-time to determine the location of assets, there are limitations to its reach inside buildings and underground storehouses.

With the use of common network tools like LAN, WIFI, and IPS; access points can be positioned at various intervals within a building to mitigate the issues faced with GPS devices. An IPS can then be deployed at low cost using wireless routers and their perspective network protocols to capture data. Wireless routers in the past have been fairly large and bulky (and non-discreet). Through technology improvements, these devices are smaller which allows them to be more mobile and placed in locations that are more constructive for gathering data and assisting in the performance of the device.

Industries are taking advantage of this evolving technology to use with location tracking of assets within the buildings which has become known as RTLS (Real Time Location Systems). Using data consumed from devices; data science models can be built to help predict the coordinates of an asset based on the signal strength from a device to the access point.

The model can then predict the location of a new unknown device based on the detected signals. This case study will explore the use of the K-Nearest-Neighbor (KNN) technique to predict X and Y coordinates of a device based data supplied on the signal strength and the access point(s) measured from the network protocols.



Data

Data Source

This case study will enlist the help of two datasets. One titled "offline" and one titled "online". The datasets were obtained from a book Nolan, D. and Temple Lang, D. (2015). *Data Science in R*. Chapman and Hall.

The data is stored in a text file format (txt) which consists of 151,392 records. Included in those numbers are 5,312 lines of comments. The raw data files contain information obtained from 6 access points (AP) placed at various locations on the first (1) floor of an office building. The "offline" data contains attributes for 166 hand-held devices that were spaced 1 meter apart.

The "online" data is comprised of 60 hand-held devices placed in random locations on the first (1) floor. To delimit the records in the dataset, a semi-colon (;) was used and the key values are defined with an equal symbol (=).

Data Analysis

In exploring the elements of the data, values such as position (POS), are separated by commas; complicating the variable exploration. The typical approach would encompass using R; applying a function to process each line into a tabular record consisting of the following attributes:

- time
- scanMAC
- address
- position
- orientation
- signal data

Given that this would be extremely time consuming in R, the project was completed using Python with Pandas to take into account the labor intensive formatting required for such a large data set and the operations to delimit them. In addition, the data is in a nested format which required the attributes of position and time to be parsed out for each of the observations. Each characteristic and position were separated from the row and then re-joined into a data-frame. Given there are no data points for position z (pos_z), it was removed.

The large amount of data required an approach to reduce the amount of orientation angles that were randomly used in the capture. Four (4) of the six (6) angles were chosen and rounded up to capture a cluster for the study. The data points not specifically belonging to the chosen angles, will be ignored.

Exploratory Data Analysis

The raw data files were processed by manually inspecting the spacing between each semi-colon (;) delimited element. Each subsection of the original file's rows had differing spacing between the labels and values which required further manipulation. In addition, varied rows of the original files were comment lines as mentioned in the introduction. These comment lines do not contain relevant data and thus were excluded.

Custom loops were developed in an effort to iterate through the row so that the spacing would allow the function to segregate specific values; focusing on entries with device_type 3. To reduce the number of attributes used for the data; each MAC signal and its strength were separated into separate rows. The MAC address could then be used as a categorical variable to assist in the KNN analysis. The manipulated data was then exported into CSV files for ease of use across multiple machines.

The final file that was manipulated contained around 90,000 entries across 8 attributes. The MAC addresses with the most entries were determined to be c0 and cd. These 2 values together comprised about 30% of the data. The average (x, y) coordinates of all the entries in the dataset were around (14,6); which were focused on the middle of the entire recorded area in the floor plan.

Figure 1: Raw View of Manipulated Data (first 5 rows)

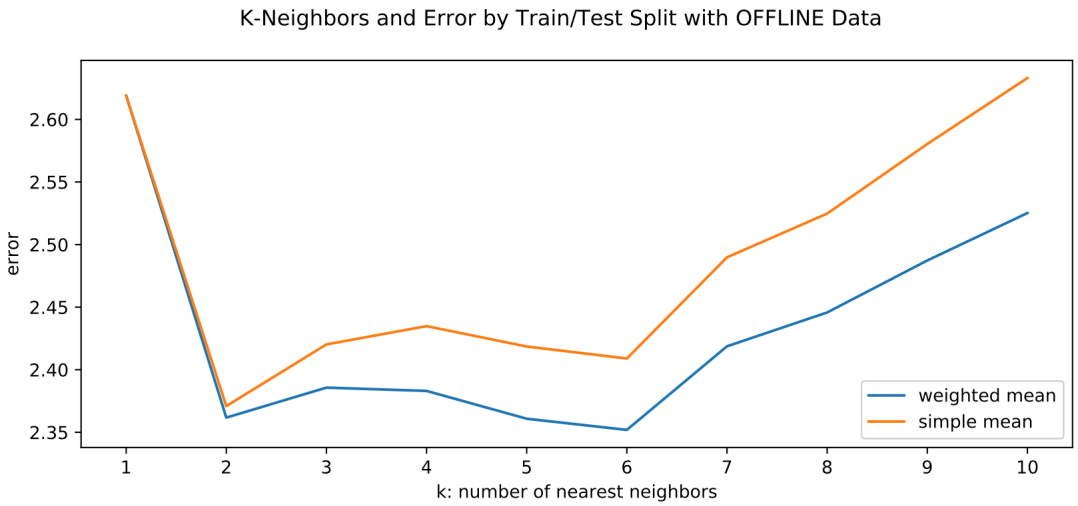
	raw_time	pos_x	pos_y	orientation	mac	signal	channel	device_type
0	2006-02-11 21:14:37.303	0.0	0.05	135.0	00:14:bf:b1:97:8a	-38	2437000000	3
0	2006-02-11 21:14:37.303	0.0	0.05	135.0	00:14:bf:b1:97:90	-56	2427000000	3
0	2006-02-11 21:14:37.303	0.0	0.05	135.0	00:0f:a3:39:e1:c0	-53	2462000000	3
0	2006-02-11 21:14:37.303	0.0	0.05	135.0	00:14:bf:b1:97:8d	-65	2442000000	3
0	2006-02-11 21:14:37.303	0.0	0.05	135.0	00:14:bf:b1:97:81	-65	2422000000	3

Method

The primary metric used for evaluating the models was error_rate. This was calculated using a comparison of the Manhattan distance between the test and train coordinates of each entry. The offline data was then used to identify the ideal n value for both the weighted and simple mean K-Nearest-Neighbors (KNN).

For the simple mean KNN; n=2 had the lowest error value. When exploring the weighted mean KNN, n=6 had the lowest error value.

Figure 2: KNN and Error for Train/Test Split



The offline dataset was also used to establish which combination of the 3 device angle orientations were most optimal for the model. Custom nested functions were used to process various iterations to determine the best angles for the model. Both the offline and online datasets were split into groupings that included MAC labels “c0”, “cd”, or both. The lowest error of parameters for the weighted KNN was determined to be k=2 and c0.

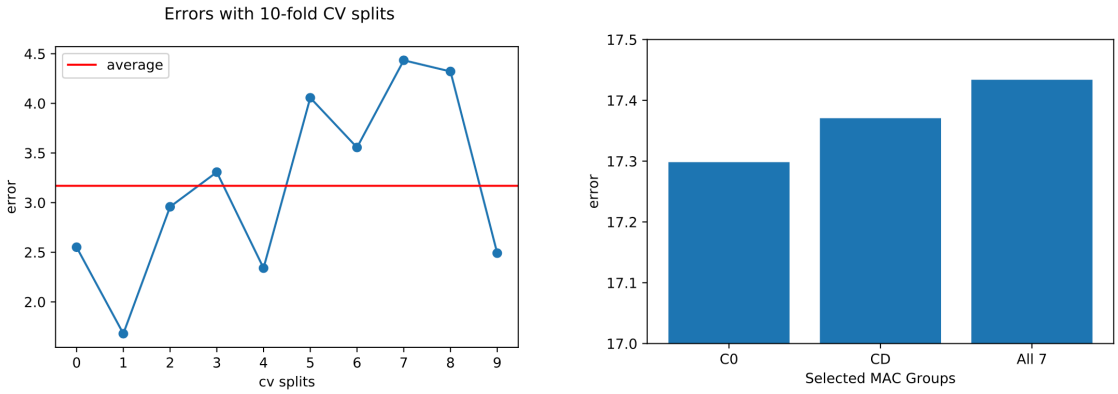
The datasets were analyzed with out-of-the-box libraries (NumPy and SciPy) to allow for quick experimentation with hyper-parameter tuning. Combinations such as Manhattan Distance with Simple Means or Euclidean Distance with Weighted Means were used to deduce the nearest neighbors.

Results

The low sample size of the online dataset (only 60 entries) may have contributed to the low success rate; as normal distribution in signal strength and other variables may not have been observed.

In an effort to confirm that the error rates lie within 1 standard deviation of the mean error, a cross-validation (10-fold) was employed.

Figure 3: Errors for Cross-Validation and Selected MAC Groups10-fold) was employed.

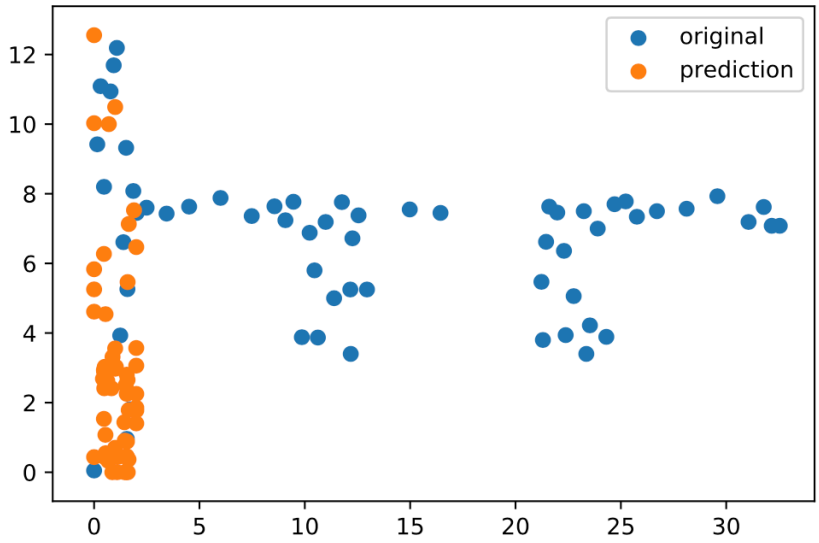


In the case study described in the book, the error rates were around 20 for the selected MAC groups. It could be deduced that the error rate achieved in this analysis is a bit stronger, but still unlikely to be used for real-world prediction modeling. The error rate of 17 doesn't give proper values for location prediction.

Conclusion

While the model produced predictions for the x, y coordinates in the held-held devices, they were not strong enough to use in real-world prediction models. It appears that the prediction intervals represented in graphical format do an injustice to the spread of devices along the x-axis of the floor plan.

Figure 4: Original vs. Predicted X, Y Values



If more time could be afforded to the study, exploration into weighting the x-axis coordinates when calculating the distances could be beneficial. The distribution signals can be different when measured between the offline/online datasets; complicating matters further.

A normalizer or regularizer function could be written before applying the "find neighbor" function to smooth out the x-axis coordinates to be more realistic in their prediction. Obviously the prediction coordinates above are not indicative of the original data points (they are skewed to the 0-5 range).

Recommendation

As the understanding of technology and applications are explored, it appears that this method is archaic. Firms that seek to track assets would be better served with the newer RFID technology. The tags are small, can be made to be undetected and are managed virtually. This makes them much more cost-effective and reliable on their tracking of assets.

Appendix

As with any technology; the system required to generate meaningful data, cleaning of the data, as well as the models that were implemented require various software packages. The raw "code" used to explore the methods detailed above can be found in the following libraries:

- Data Processing
- Analysis and Modeling in Python
- Location Prediction for KNN