Coursera Capstone Report: Farmer’s Market Locator

Introduction where you discuss the business problem and who would be interested in this project.

Choosing the right location for a farmer’s market can be the key to its success or failure. If organizers had a tool that would help them vet locations, it would allow them to avoid costly mistakes due to locating the market in the wrong part of the city.

Data where you describe the data that will be used to solve the problem and the source of the data.

The data we will be using comes primarily from Kaggle and Foursquare. We will also need to generate some farmersMarket=negative datapoints ourselves.

Here is a link to registered farmers market locations from Kaggle:

https://www.kaggle.com/madeleineferguson/farmers-markets-in-the-united-states

This data on its own is not enough to train a model as each row is dedicated to an existing farmers market. To train the model properly we will have to generate additional data points so the model can consider how the features from unsuitable locations will affect it.

To simplify our (farmers market = false) datapoint creation process, we will create GPS coordinates 3, 6, 20, and 60 miles to the south of every farmers market in the data set and assume that they will not contain farmers markets.

After generating the new data points, we should have more than 35,000 rows to work with.

The combined Kaggle and generated data would exceed 35,000 rows if we utilized all the available Kaggle farmer’s market data. However, Foursquare has an API limit of 950 calls per day. Thus, to pull our data in a timely fashion we will need to limit the number of farmer’s markets we analyze to approximately 190. Conveniently Colorado, has 161 registered farmers markets in the dataset which places us inside the daily call limit with some wiggle room for testing.

Foursquare will be used to pull a list of the restaurants and amenities within 1.5 miles of existing farmers markets. We will also pull the same information for the coordinates we generated. The final dataset should look something like this:

Latitude	Longitude	# Indian Rest	# Italian Rest	# Parks	[…]	Farmer's Market Present
-105.073	40.395401	6	2	1	…	True
-106.817	39.1894	0	0	8	…	False

After the data has been prepped, we will use it to train a KNeighbors classifier.

The program will take an input latitude and longitude, pull the restaurants and amenities around the coordinates, run the features through our KNeighbors classifier, and then determine whether the location is similar enough to existing registered farmer's markets to be a good selection.

To improve the utility of our program, we will create a map of potential farmer's market locations for the city of Denver. To do this we would overlay a grid of GPS coordinates and run our algorithm against each coordinate in that 20 x 20 grid.

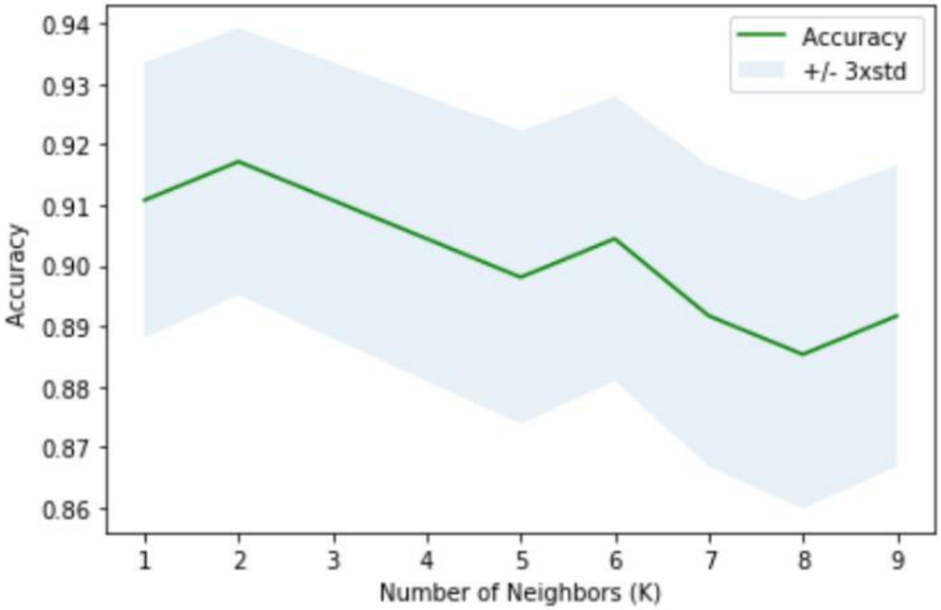
Methodology section which represents the main component of the report where you discuss and describe any exploratory data analysis that you did, any inferential statistical testing that you performed, if any, and what machine learnings were used and why.

This program did not really require much exploratory analysis. Instead I was able to jump directly into feature engineering with the bulk of the code dealing with massaging my dataframe columns into just the rights shapes. I chose the k-nearest neighbors' algorithm because I was predicting a category, utilizing labeled data, with less than 100,000 samples, and my data was entirely numeric. I also considered using a Linear SVC but opted not to as I was not familiar enough with that algorithm. One of the advantages to using the KNeighbors algorithm was that I felt it handled edge cases well for this particular problem set.

Program Workflow:

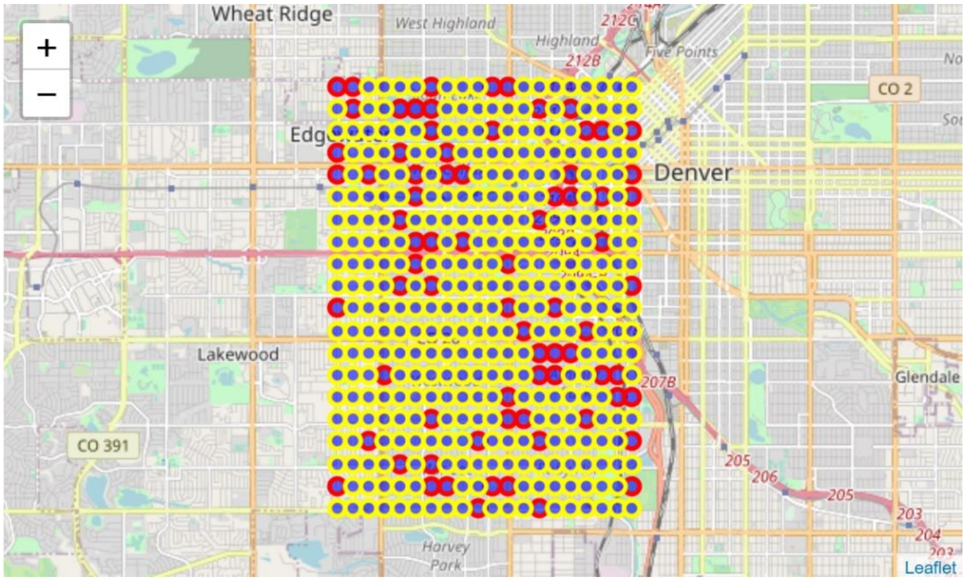
1. Import libraries
2. Load farmer’s market data
3. Transform farmer’s market data
4. Write function to add and subtract distance from geo coordinates
5. Generate null datapoints for farmer’s markets
6. Plot lat/lon datapoints on a map
7. Pull Foursquare restaurants & amenities for each geo coordinate
8. Merge farmer’s market & Foursquare data
9. Split training & test data
10. Test k values for KNeighbors classifier
11. Train KNeighbors classifier
12. Create a grid of lat/lon coordinates over Denver
13. Plot test coordinates on map
14. Pull Foursquare data for each lat/lon in the grid
15. Run each gridpoint against the KNeighbors model
16. Plot a map of Denver with predicted farmer's market locations from gridpoints
17. Closing thoughts

Results section where you discuss the results.



Utilizing two nearest neighbors, my training set accuracy was 95% and my test set accuracy was 91%.

After generating a grid of coordinates over Denver, 76 of 400 coordinates were found to be locations with a similar local business makeup to existing farmers markets. In simpler terms we found 76 potential farmer's market locations.



The red dots indicate matches, and the yellow dots indicate the rest of the tested points which came back negative.

Discussion section where you discuss any observations you noted and any recommendations you can make based on the results.

Based on close analysis of the map, it seems like farmers markets are placed in locations with proximity to a park, campus, or body of water. Farmer’s markets also seem more frequent in areas with large parking lots. Farmer’s markets also seem more prevalent off large intersections. Lastly streets proximity to mid-range restaurants seem to make the presence of a farmer’s markets more likely.

I would say the algorithm shows promise, but the dataset should be expanded much larger. After upgrading my FourSquare accounts API limits, the application ran more reliably, but would still have occasional data hang-ups requiring me to rerun expensive api calls. The reason I did not expand my training and test points after I finished the program was that I did not feel confident enough in the stability of the FourSquare API. I might have written a program to handle the data errors more gracefully, but that would’ve required stretched the project outside of my time constraints.

Conclusion section where you conclude the report.

The map shows 76 potential locations for farmers markets based on the similarity of surrounding amenities to existing farmer's markets. This number is higher than I would have imagined as I only created a 400-point grid. To pull the data in a timely fashion and deal with the API restrictions I had to artificially limit my training and testing data. There is no doubt expanding my dataset would yield a more selective ML algorithm. One point I am sure is causing false positives is the lack of negative datapoints from densely populated urban areas. If I were to make one modification with my existing data it would be to add a grid of closely spaced negative datapoints over Denver, away from existing farmers markets.