FARMER'S MARKET LOCATOR

Coursera Data Science Professional Capstone Project

How do you choose the right location for a new farmer's market?

- 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.
- We can train a machine learning model to predict good locations for markets based on the shops around existing markets.

Data

- Kaggle: Dataset of farmers markets in the US with latitude and longitude
- Self-generated: List of additional coordinates reflecting areas where there are no farmer markets.
- Foursquare: Dataset of the top 100 nearby amenities for any given coordinate

Data: Kaggle

	id	name	lon	lat	farmersMarket
37	1006234	4th Street Farmers Market	-105.07300	40.395401	1.0
114	1004070	Alamosa Farmers Market	-105.86523	37.468361	1.0
179	1001367	American National Bank Downtown Farmers Market...	-108.56400	39.068199	1.0
289	1005081	Arvada Farmers Market	-105.08145	39.800137	1.0
290	1009285	Arvada Five Parks Farmers Market	-105.15500	39.848801	1.0

- After removing extraneous columns this is what was left
- Note the ids for these original datapoints all start with 1

Data: Self Generated

	id	name	lon	lat	farmersMarket
0	2012468	4th Street Farmers Market	-105.07300	40.351984	0.0
1	2008140	Alamosa Farmers Market	-105.86523	37.424944	0.0
2	2002734	American National Bank Downtown Farmers Market...	-108.56400	39.024782	0.0
3	2010162	Arvada Farmers Market	-105.08145	39.756720	0.0
4	2018570	Arvada Five Parks Farmers Market	-105.15500	39.805384	0.0

- I generated negative datapoints by offsetting coordinates from the existing markets.
- To generate these negative datapoints I copied the Kaggle dataset four times. On each copy I transformed the latitude by a set distance. I calculated latitude offsets for 3, 6, 20, and 60 miles. This made the negative datapoint creation simple.
- I multiplied the id number by an incrementing value to generate unique ids for the copies

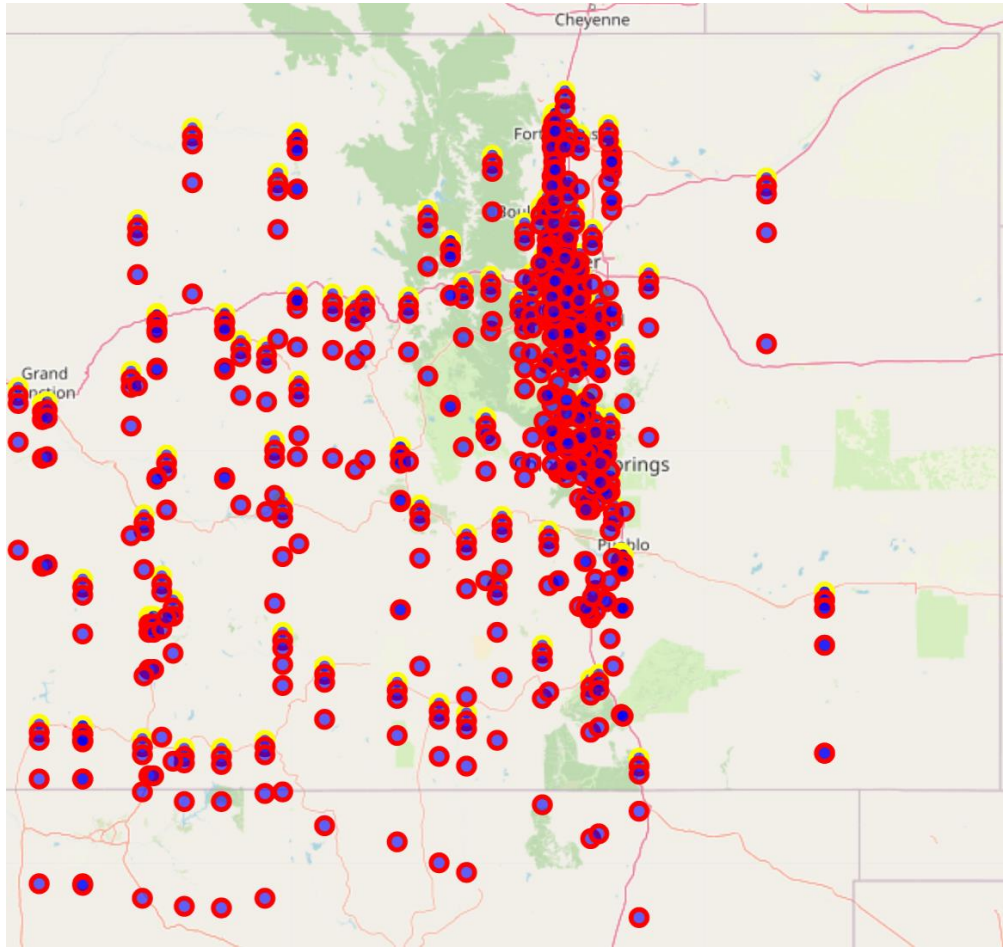
Data: Foursquare

	uniqueId	ATM	Accessories Store	Adult Boutique	Advertising Agency	Airport	American Restaurant	Animal Shelter	Antique Shop	Arcade	...	Vietnamese Restaurant	Vineyard	Warehouse Store	Whi
0	1000015	0.00000	0.0	0.0	0.0	0.0	0.107143	0.0	0.0	0.00000	...	0.0	0.0	0.0	0.00
1	1000133	0.01087	0.0	0.0	0.0	0.0	0.054348	0.0	0.0	0.01087	...	0.0	0.0	0.0	0.01
2	1000315	0.00000	0.0	0.0	0.0	0.0	0.090909	0.0	0.0	0.00000	...	0.0	0.0	0.0	0.00
3	1000417	0.00000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.00000	...	0.0	0.0	0.0	0.00
4	1000418	0.00000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.00000	...	0.0	0.0	0.0	0.00

- The data I pulled from Foursquare got transformed into this dataframe tying the coordinate ID to a fingerprint of the shops around it.
- The Foursquare API broke often so I limited how much data I had to pull from it.

Data: Kaggle + Self Generated on Map

- Yellow dots represent existing farmer's markets
- Red dots represented the negative datapoint I generated using coordinate offsets
- It is skewed along the highway, which was efficient as it captured more of Colorado's urban areas.
- Skewing points left or right would not have resulted in a benefit as they are relatively unpopulated areas.

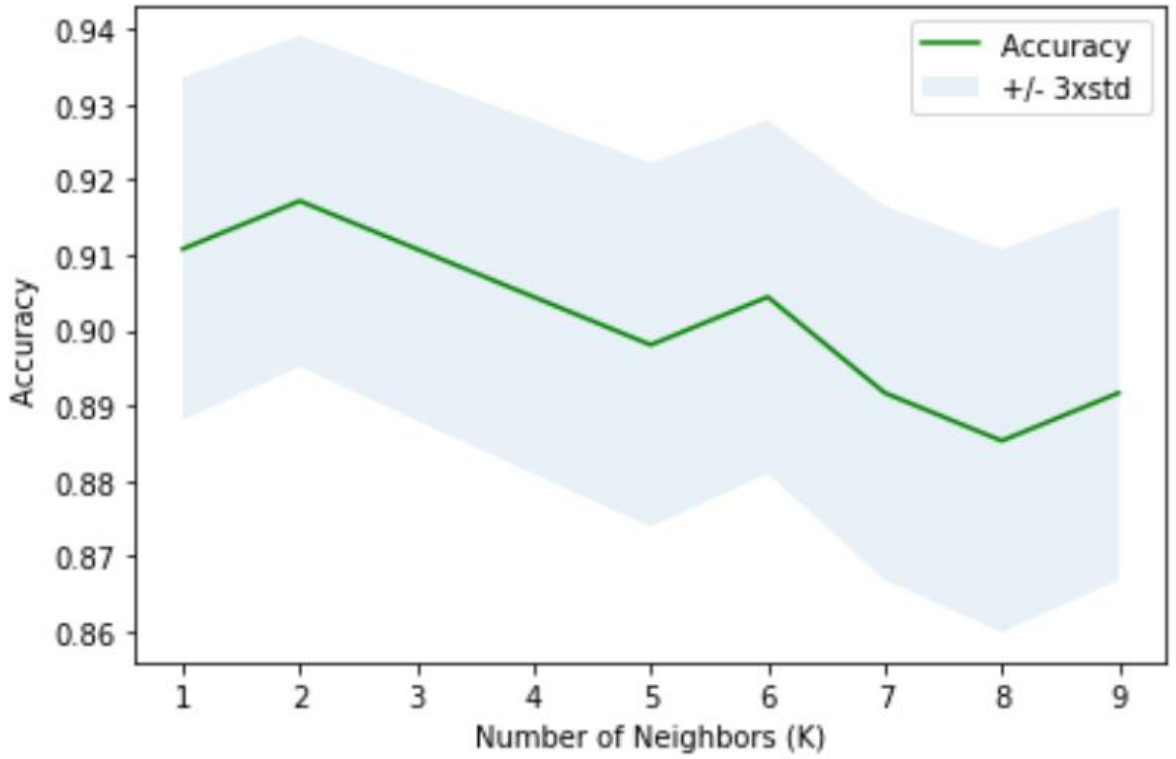


K Nearest Neighbor

- I chose to use the K Nearest Neighbor algorhtim because I was:
  - Predicting a category
  - Using labeled data
  - With less than 100,000 samples
  - Data was numeric, not text

- I also considered a Linear SVC but opted not to go this direction as I didn't have experience with this algorithm

- I thought the KNeighbors algorithm would handle edge cases well for this problem set

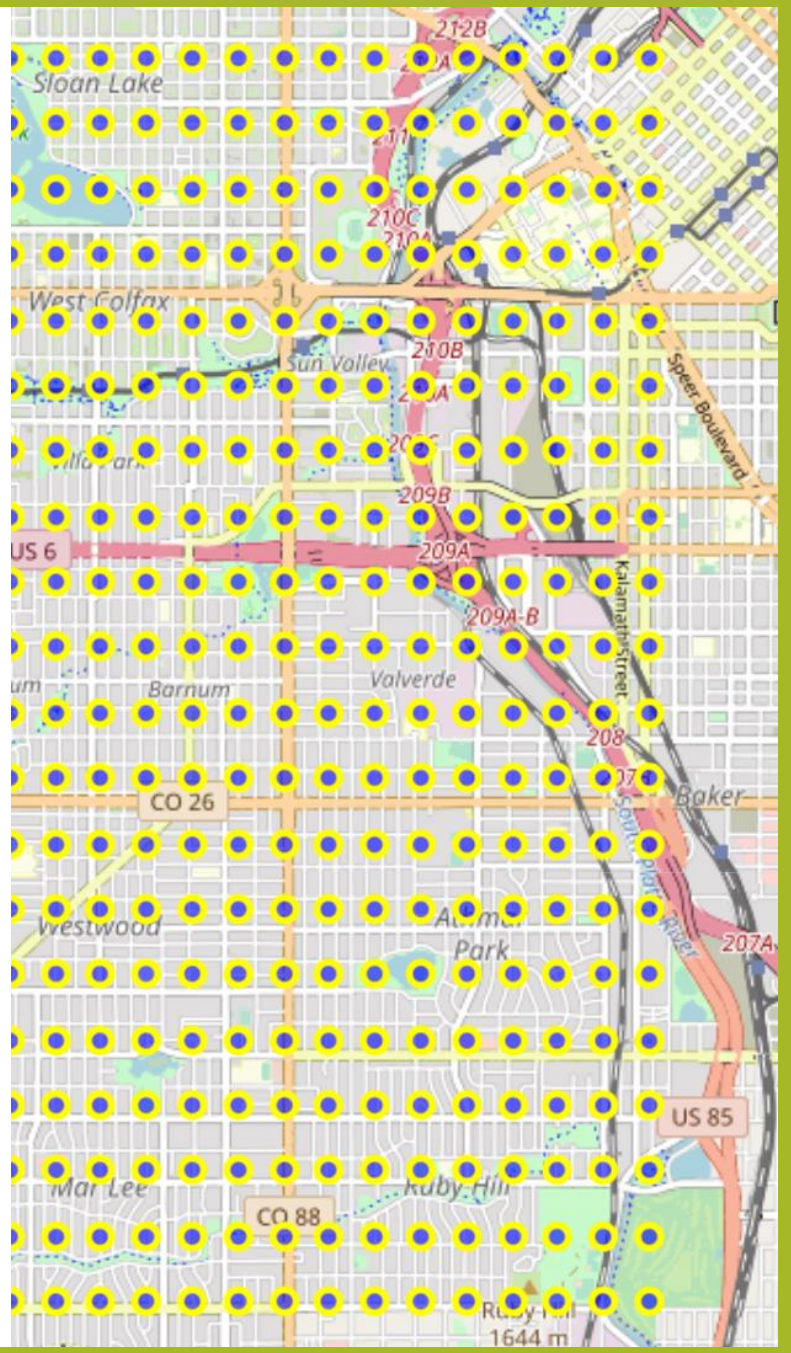


K Nearest Neighbor: Selecting K Value

- I selected a K value of 2 as that appeared to have the highest accuracy when testing the model
- I was happy with the accuracy of the program, although I know exactly where it's faults would materialize.

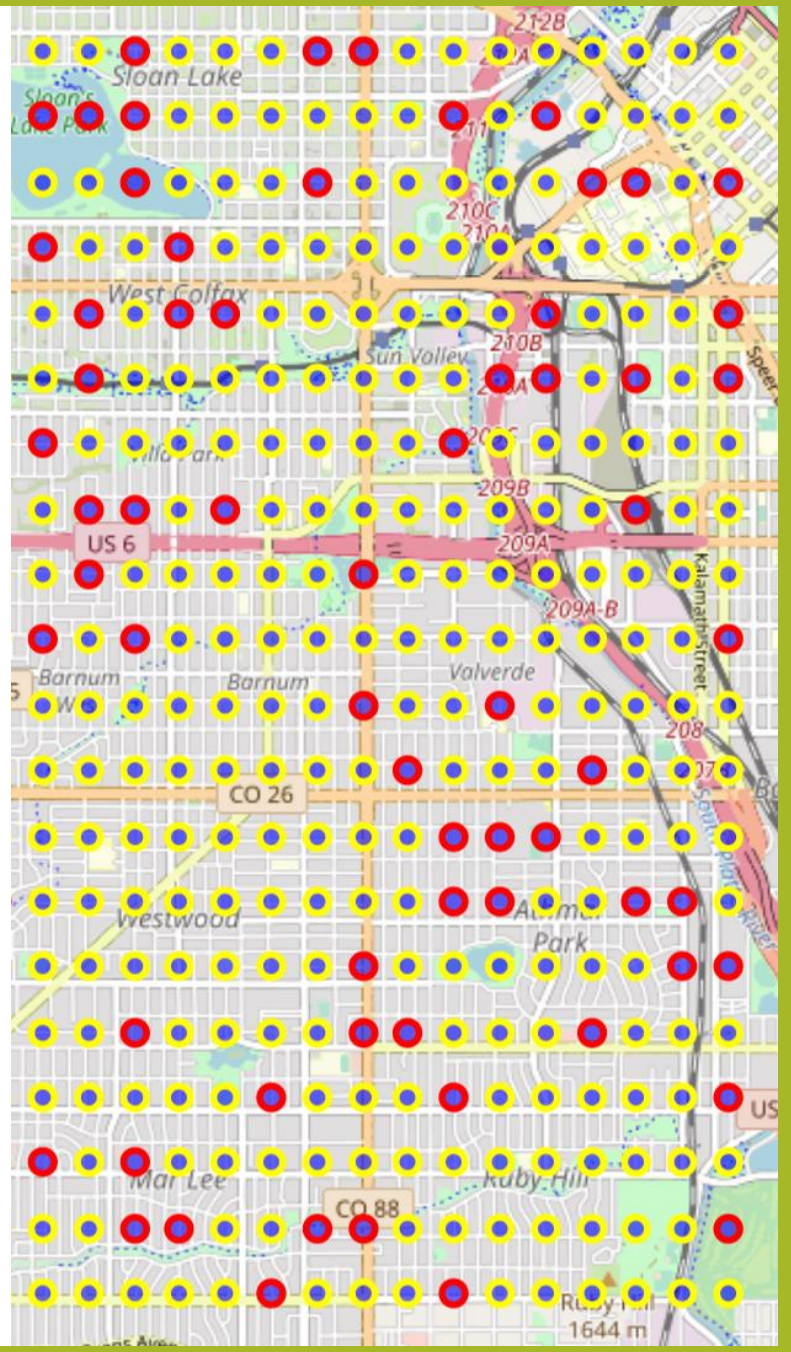
Prediction: Generating coordinate grid

- In order to test the algorithm I generated a coordinate grid, that I would then use to feed into the program



Prediction: Results

- Surprisingly the 19% of the tested coordinates came back as potential locations for a farmer's markets
- This seems high and is likely due to not having enough negative datapoints from urban areas.
- That said there does seem to be a preference for proximity to parks, intersections, parking lots, and local attractions which means we are on the right track.



Discussion

The results show promise!

I wouldn't base a decision to locate a farmer's market based on this program yet. To feel confident I would need to greatly expand the dataset, and I would probably also want to add features with economic and demographic data.

The major limiting factor was the stability of the Foursquare API.

If I was to extend this project, I would work on writing code to handle API errors more gracefully, and store data incrementally.

This approach could be used to locating other venue types, such as offices, coffee shops, or restaurants.