

# CAPSTONE PROJECT: FINAL REPORT

## LOST AND FOUND PETS STATISTICS AND SEGMENTATION

### 1. PURPOSE

---

### 2. INTRODUCTION

---

Missed pets have become a severe problem in many cities. Pets owners usually take a long time to find them or even they may never find their pets because, for example, these pets might be moved far from where they used to live. On the other hand, ONG's have high difficulty to rescue rejected pets, so these ONGs are not often alerted about them or when are these pets have already moved to another place.

The mobile phones have already become part of the day-to-day of people's lives, who might provide using their mobile phones the geographic location and a photo of these missed or rejected pets. In its turn, pets owners and ONGs could search and visualize the geographic location of missed or rejected pets including points of references such as cafes and restaurants. Furthermore, governmental agencies could visualize and compare areas of the city with the highest incidence of missed ou rejected and could segment areas to uncover patterns which could guide marketing campaigns to mitigate the problem.

### 3. DATA DESCRIPTION AND ACQUISITION

---

This prototype will make use of the following data sources:

#### Animal Services of The City of Toronto

The **Stray Animals Report** provide by The Animal Services of The City of Toronto displays stray animals (cats and dogs) received in the last 5 days. The report data will be scraped from <https://www.toronto.ca> and contains the following information:

- **Category:** Cat or Dog
- **Date**
- **Breed**
- **Approximate Age**

- **Sex:** Male or Female
- **Colour**
- **Receiving Shelter**
- **Animal ID Number**
- **Crossing Intersection**

The **Localisation of Receiving Shelters** data provided by The Animal Services of City of Toronto will be scraped from <https://www.toronto.ca/community-people/animals-pets/animal-shelters/> and contains the following information:

- **Title**
- **Address**

**Toronto Venues nearby Crossing Intersections from FourSquare API (FourSquare website: [www.foursquare.com](http://www.foursquare.com))**

The FourSquare API will be used to explore neighborhoods in **Crossing Intersections** and **Receiving Shelters Localisation** in Toronto. The Foursquare explore function will be used to get the most common venue categories in each neighborhood, and then use this feature to group the neighborhoods into clusters. The following information are retrieved on the first query:

- **Venue ID**
- **Venue Name**
- **Coordinates:** Latitude and Longitude
- **Category Name**

## 4. METODOLOGY

### Lost and Found Pets

The data source contains the information about stray animals received in the last 5 days by The Animal Services of the City of Toronto.

**1. Data Cleaning** The report is available in two HTML tables (cats and dogs). These table contains some inconsistent entries and needs some cleanup.

The following activities were performed:

- Drop/ignore cells with missing crossing intersections data
- Fix cells with crossing intersections wrong format.

- Separate crossing intersections fields in street 1 and street 2.

	date	breed	age	se
0	2019-01-09	DOMESTIC SH		Ma
1	2019-01-09	DOMESTIC SH		Unknow
2	2019-01-09	DOMESTIC SH	5M	Ma
3	2019-01-09	DOMESTIC SH	2Y	Fema
4	2019-01-12	DOMESTIC SH	3Y	Ma
5	2019-01-12	DOMESTIC SH		Unknow
6	2019-01-12	DOMESTIC SH		Ma
7	2019-01-13	DOMESTIC SH		Fema
8	2019-01-09	SHIH TZU	5Y	Ma
9	2019-01-09	SIBERIAN HUSKY		Neutere Ma

**Post processed sample Lost and Found pets table.**

**2. Localisation of crossing intersections** The Geocoder Service (<https://geocoder.api.here.com>) was used to find latitude and longitude of crossing intersections. These geographical coordinates will be used to search FourSquare API location data.

**The Python code used to retrieve geographical coordinates for crossing intersections.**

```
“python def getcrossinterseclocalization(pets): url = 'https://geocoder.api.here.com/6.2/geocode.json?city=toronto&apiid='79foQR1GPJRvsWDGB0UI' apicode = 'E5YKLSl_029hf-ipUIPFfQ'”
```

```
for row in pets.itertuples():
    address = url.format('Toronto'
                        , row.cross_intersec_st1
                        , row.cross_intersec_st2
                        , '79foQR1GPJRvsWDGB0UI'
                        , 'E5YKLSl_029hf-ipUIPFfQ')
    response = requests.get(address).json()
    try:
        localization = json_normalize(response['Response']['View'][0]['Result'][0]['Location'])
        pets.loc[row.Index, 'cross_intersec_latitude'] = localization.loc[0, 'DisplayPosition']
        pets.loc[row.Index, 'cross_intersec_longitude'] = localization.loc[0, 'DisplayPosition']
```

```

except Exception as e:
    print('Crossing intersection {}/{} was not found in geocode database: {}! '.format(
        row.cross_intersec_st1
        , row.cross_intersec_st2
        , str(e)))

return(pets)

```

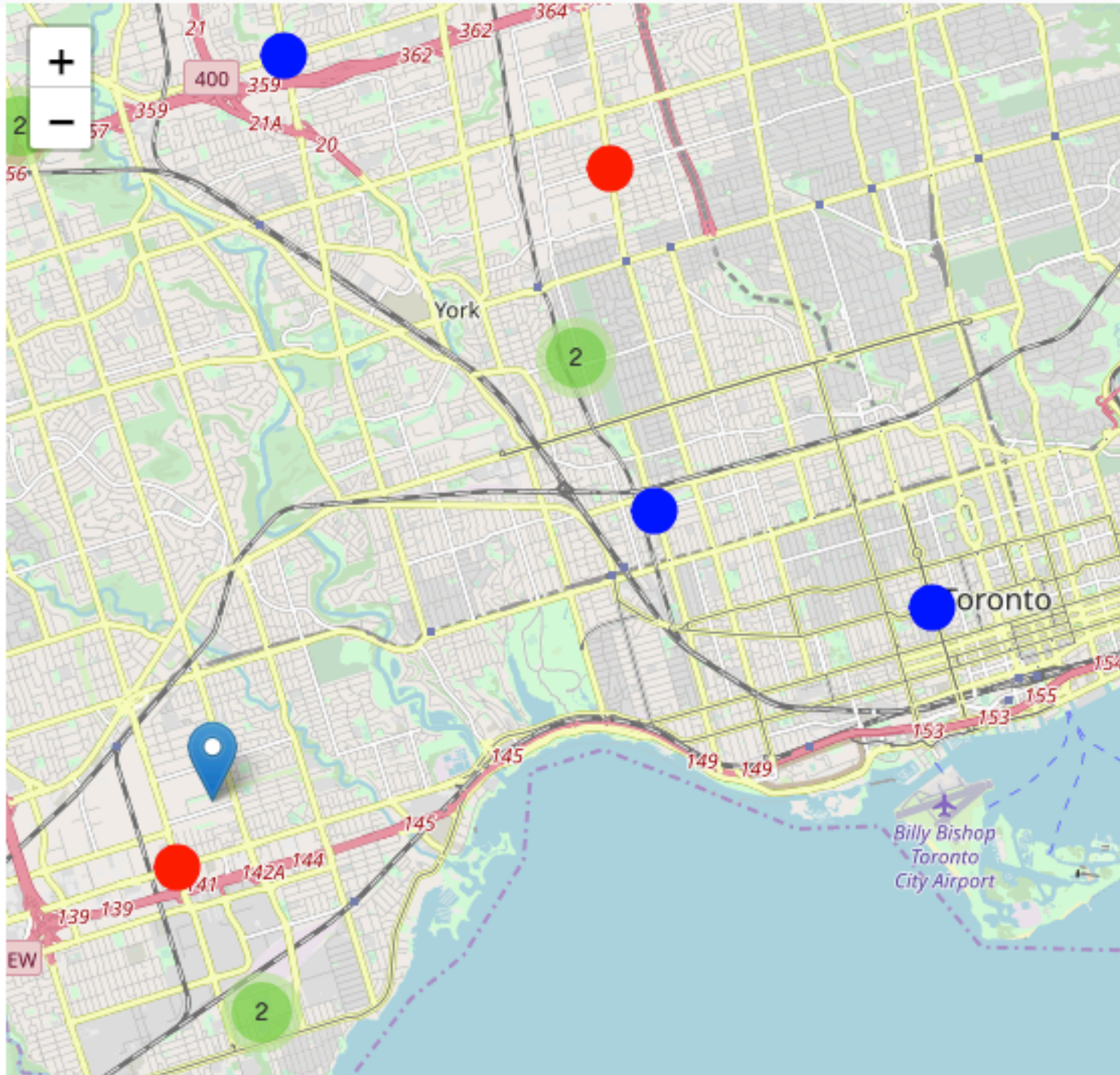
““

Post processed sample Lost and Found pets table merged with geographical coordinates.

	id	crossing_intersections	cross_intersec_st1	cross_intersec_st2	cross_inte
0	A824678	THE QUEENSWAY/KIPLING	THE QUEENSWAY	KIPLING	
1	A824739	ISLINGTON BIRMINGHAM	ISLINGTON	BIRMINGHAM	
2	A824752	REXDALE/ISLINGTON	REXDALE	ISLINGTON	
3	A824753	REXDALE/ISLINGTON	REXDALE	ISLINGTON	
4	A824906	DANFORTH/VICTORIA PARK	DANFORTH	VICTORIA PARK	
5	A824912	KIPLING/401	KIPLING	401	
6	A824932	DUFFERIN/GLENCAIRN	DUFFERIN	GLENCAIRN	
7	A824940	DUFFERIN/ROGERS	DUFFERIN	ROGERS	
8	A824737	SPADINA DUNDAS	SPADINA	DUNDAS	
9	A824755	LANSDOWNE AVE	LANSDOWNE	AVE	

## 5. SEGMENTATION AND CLUSTERING

Post processed sample Lost and Found pets table merged with geographical coordinates.



## 6. DISCUSSION AND CONCLUSION