

Introduction

Problem

About a quarter million people are moving to Colorado every year. Many of these people are looking to set up new business and may not know what city they would want to set up their business. Specifically in Denver since it is the biggest city in Colorado. This is important so people who are starting up their business are in a great location with a lot of demand which would bring in customers and make a lot of money.

Data

The data I will be using are zip codes of all the neighborhoods in Denver. I will be joining these with other zip code data sets that are specific to each neighborhood. I will be using this data to find optimal neighborhoods to build a specific business in and what areas in those neighborhoods are best. I will be using foursquare to find information on each cities and see what kind of business are in each zip codes in order to provide feedback on what businesses would be ideal in each area. Some of the features we will get from this data are the different types of business in an area and how many of them there are in that area. This can be a clear indicator of what business would be best in what areas.

```
In [105]: # importing packages
import pandas as pd
import numpy as np
from geopy.geocoders import Nominatim
import requests
from pandas.io.json import json_normalize
import matplotlib.cm as cm
import matplotlib.colors as colors
from sklearn.cluster import KMeans
import folium
import seaborn as sns
from geopy.geocoders import Nominatim
import sklearn
from sklearn.neighbors import NearestNeighbors
from sklearn.linear_model import LogisticRegression
```

```
In [2]: # Importing data

df_nb = pd.read_csv ('/Users/jairgalindoflores/Documents/GitHub/IBM-Capstone/De\
nver_neighborhoods_data.csv', index_col=0)

df_zip = pd.read_csv ('/Users/jairgalindoflores/Documents/GitHub/IBM-Capstone/u\
sziips.csv')
```

```
In [43]: # joining data

df_merged= pd.merge(df_nb, df_zip, on='zip')
df = df_merged[['Neighborhood', 'zip', 'lat', 'lng']]
df.head(10)
```

Out[43]: (78, 4)

Exploratory Data Analysis

Understanding data

```
In [4]: # getting the shape of the dataframe
print('The data frame has', df.shape[0], 'neighborhoods and a total of',
df.shape[1], 'columns.')
```

The data frame has 78 neighborhoods and a total of 4 columns.

```
In [5]: # data types of the columns
df.dtypes
```

Out[5]: Neighborhood object
zip int64
lat float64
lng float64
dtype: object

```
In [6]: df.nunique(axis=0)
```

Out[6]: Neighborhood 78
zip 32
lat 32
lng 32
dtype: int64

What we can see from this function is that a lot of the neighborhoods are very close to each other. What we might have to do is remove or merge together some of the neighborhoods that have data that are the same.

```
In [7]: # get the coordinatesg for Denver

geolocator = Nominatim(user_agent='co_explorer')
location = geolocator.geocode('Denver, Colorado')
lat = location.latitude
lng = location.longitude
print('The geographical coordinate of Denver is ( ) ( ).'.format(
location.latitude, location.longitude))
```

The geographical coordinate of Denver is 39.7392364 -104.984623.

```
In [8]: # create map of Toronto using latitude and longitude values
denver_map = folium.Map(location=[lat, lng], zoom_start=10)
# add markers to map
for lat, lng, neighborhood in zip(df[['lat']], df[['lng']], df[['Neighborhood']]):
    label = '{0}'.format(neighborhood)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(denver_map)
```



```
In [9]: # Define Foursquare Credentials and Version
CLIENT_ID = "0BKDKQNCVTRN2LTHOWDXJ4FDSHNE3KMKCZW5H3GXTD5KU"
CLIENT_SECRET = "53QE3SDM52XLB83EUB4F9SFVRUTNSOQAF3KCNJOFV3AL6QR4"
VERSION = "20180603"
LIMIT = 100
```

```
In [10]: # function that extracts the category of the venue
def get_category_type(row):
    try:
        categories_list = row['categories']
    except:
        categories_list = row['venue.categories']

    if len(categories_list) == 0:
        return None
    else:
        return categories_list[0]['name']
```

```
In [11]: # function to apply to individual neighborhoods in toronto
def getNearbyVenues(names, latitudes, longitudes, radius=500):

    venues_list=[]
    for name, lat, lng in zip(names, latitudes, longitudes):
        print(name)

        # create the API request URL
        url = 'https://api.foursquare.com/v2/venues/explore?client_id={}&client_secret={}&v={}&ll={}&radius={}&limit={}'.format(
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            lat,
            lng,
            radius,
            LIMIT)

        # make the GET request
        results = requests.get(url).json()["response"]["groups"][0]["items"]

        # return only relevant information for each nearby venue
        venues_list.append(((
            name,
            lat,
            lng,
            v['venue']['name'],
            v['venue']['location']['lat'],
            v['venue']['location']['lng'],
            v['venue']['categories'][0]['name']) for v in results))

    nearby_venues = pd.DataFrame([item for venue_list in venues_list for item in venue_list])
    nearby_venues.columns = ['Neighborhood',
                            'Neighborhood Latitude',
                            'Neighborhood Longitude',
                            'Venue',
                            'Venue Latitude',
                            'Venue Longitude',
                            'Venue Category']

    return(nearby_venues)
```

```
In [12]: denver_venues = getNearbyVenues(names=df[['Neighborhood']],
                                         latitudes=df[['lat'],
                                         longitudes=df[['lng']]
                                         )
```

Athmar Park	Overland	Valverde	Auraria	Barnum	Lincoln Park	Platt Park	Sun Valley	Villa Park	West Colfax	Baker	Belcaro	Cherry Creek	Washington Park	Washington Park West	Barnum West	Harvey Park	Mar Lee	Westwood	Bear Valley	Berkeley	Highland	Regis	Sloan Lake	Capitol Hill	Speer	North Capitol Hill	City Park West	Country Club	CBD (downtown)	Civic Center	Union Station (Lodo)	Central Park	Chaffee Park	Ruby Hill	Cheesman Park	Congress Park	City Park	Clayton	Cole	Skyland	Whittier	College View / South Platte	Harvey Park South	Cory-Merrill	Rosedale	University	University Park	Welshire	DFA	Gateway / Green Valley Ranch	East Colfax	Bale	Montclair	South Park Hill	Elyria-Shumans	Globeville	Five Points	Fort Logan	Goldsmith	University Hills	Hamden	Southmoor Park	Hamden South	Hilltop	Virginia Village	Washington Virginia Vale	Indian Creek	Jefferson Park	Sunshine	West Highland	Kennedy	Lowry Field	Marston	Montbello	Northeast Park Hill	North Park Hill	Windsor
-------------	----------	----------	---------	--------	--------------	------------	------------	------------	-------------	-------	---------	--------------	-----------------	----------------------	-------------	-------------	---------	----------	-------------	----------	----------	-------	------------	--------------	-------	--------------------	----------------	--------------	----------------	--------------	----------------------	--------------	--------------	-----------	---------------	---------------	-----------	---------	------	---------	----------	-----------------------------	-------------------	--------------	----------	------------	-----------------	----------	-----	------------------------------	-------------	------	-----------	-----------------	----------------	------------	-------------	------------	-----------	------------------	--------	----------------	--------------	---------	------------------	--------------------------	--------------	----------------	----------	---------------	---------	-------------	---------	-----------	---------------------	-----------------	---------

```
In [13]: denver_venues
```

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Athmar Park	39.69617	-105.00186	Chain Reaction Brewery	39.696577	-105.001335	Brewery
1	Athmar Park	39.69617	-105.00186	Rome's Saloon	39.696839	-104.996446	Sports Bar
2	Athmar Park	39.69617	-105.00186	National Bantickade Co Of Denver Pk	39.696581	-105.000135	Clothing Store
3	Athmar Park	39.69617	-105.00186	Kroger Central Fill	39.694432	-105.001271	Pharmacy
4	Athmar Park	39.69617	-105.00186	Tony's Handy Man Services	39.696418	-105.004287	Home Service
...
1351	Windsor	39.69715	-104.88179	Breakers Cafe	39.699197	-104.883001	Bar
1352	Windsor	39.69715	-104.88179	Surfish Lake Trail	39.699125	-104.879405	Lake
1353	Windsor	39.69715	-104.88179	Windsor Lake	39.699067	-104.879317	Lake
1354	Windsor	39.69715	-104.88179	The Pool	39.694125	-104.883602	Pool
1355	Windsor	39.69715	-104.88179	Garage Service Co	39.699516	-104.877871	Construction & Landscaping

1356 rows x 7 columns

```
In [14]: denver_venues.groupby('Neighborhood').count()
```

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Athmar Park	10	10	10	10	10	10	10
Auraria	14	14	14	14	14	14	14
Baker	3	3	3	3	3	3	3
Barnum West	3	3	3	3	3	3	3
Barnum	14	14	14	14	14	14	14
...
West Colfax	14	14	14	14	14	14	14
West Highland	22	22	22	22	22	22	22
Westwood	3	3	3	3	3	3	3
Whittier	9	9	9	9	9	9	9
Windsor	5	5	5	5	5	5	5

77 rows x 6 columns

```
In [15]: print('There are {} unique categories.'.format(len(denver_venues['Venue Category'].unique())))
There are 162 unique categories.
```

```
In [32]: # one hot encoding
denver_onehot = pd.get_dummies(denver_venues[['Venue Category']], prefix="", prefix_sep='')
# add neighborhood column back to dataframe
denver_onehot['Neighborhood'] = denver_venues['Neighborhood']
# move neighborhood column to the first column
fixed_columns = [denver_onehot.columns[0]] + list(denver_onehot.columns[1:-1])
denver_onehot = denver_onehot[fixed_columns]
denver_onehot.head()
```

	Acupuncturist	Acupuncturist	Alternative Healer	American Restaurant	Antique Shop	Arcade	Art Gallery	Arts & Crafts Store	Arts & Entertainment	Asian Restaurant	...	Track	Trail	Train Station
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0

5 rows x 162 columns

```
In [17]: denver_grouped = denver_onehot.groupby('Neighborhood').mean().reset_index()
denver_grouped.head()
```

	Neighborhood	Yoga Studio	Acupuncturist	Alternative Healer	American Restaurant	Antique Shop	Arcade	Art Gallery	Arts & Crafts Store	Arts & Entertainment	...	Track	Trail	Train Station
0	Athmar Park	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0
1	Auraria	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0
2	Baker	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0
3	Barnum West	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0
4	Barnum	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0

5 rows x 162 columns

