1.Introduction

Chennai (/ˈtʃɛnaɪ/ (listen), Tamil: [tʃen:ai]), also known as **Madras** (/məˈdrɑ:s/ (listen) or /-ˈdræs/,[13] the official name until 1996), is the capital of the Indian state of Tamil Nadu. Located on the Coromandel Coast off the Bay of Bengal, it is the biggest cultural, economic and educational centre of south India. According to the 2011 Indian census, it is the sixth-most populous city and fourth-most populous urban agglomeration in India. The city together with the adjoining regions constitute the Chennai Metropolitan Area, which is the 36th-largest urban area by population in the world.[14] Chennai is among the most-visited Indian cities by foreign tourists. It was ranked the 43rd-most visited city in the world for the year 2015.[15] The Quality of Living Survey rated Chennai as the safest city in India.[16] Chennai attracts 45 percent of health tourists visiting India, and 30 to 40 percent of domestic health tourists.[17] As such, it is termed "India's health capital".[18][19]

Chennai had the third-largest expatriate population in India, at 35,000 in 2009, 82,790 in 2011 and estimated at over 100,000 by 2016.[20][21] Tourism-guide publisher Lonely Planet named Chennai as one of the top ten cities in the world to visit in 2015.[22] Chennai is ranked as a beta-level city in the Global Cities Index,[23] and was ranked the best city in India by *India Today* in the 2014 annual Indian city survey.[24][25] In 2015 Chennai was named the "hottest" city (worth visiting, and worth living in for long term) by the BBC, citing the mixture of both modern and traditional values.[26] National Geographic mentioned Chennai as the only South Asian city to feature in its 2015 "Top 10 food cities" list.[27] Chennai was also named the ninth-best cosmopolitan city in the world by Lonely Planet.[28] In October 2017, Chennai was added to the UNESCO Creative Cities Network (UCCN) list for its rich musical tradition.[29]

Problem Statement

Chennai has been the capital of Tamil Nadu. The problem is to find the right place for construting a Apartment so there will be lot of insights.As tehe city has been expanding for the past 20 years and has been the IT hub of Tamil Nadu.So Residential and Real estate has been the dominant business in this region, So the basic idea is to use clustering to group the cities based on apartment and find the right place for constructing new building

Steps to be Followed

Step1: Find the suburbs of chennai

Step2: Find lat long using geopy module

Step3: Use foursquare api to find different venue

Step4: Cluster places based on similarity using K-Means

Step5 :Conclusion

Step1: Find the suburbs of chennai

The first step is to download the dataset from wiki page
https://en.wikipedia.org/wiki/List_of_neighbourhoods_of_Chennai

using read_html and create a pandas dataframe

this results in multiple df and we are extracting the data and converting it into a list

Step2: Find lat long using geopy module

2.1 Geopy Introduction

Geopy is a Python 2 and 3 client for several popular geocoding web services.
geopy makes it easy for Python developers to locate the coordinates of addresses, cities, countries, and landmarks across the globe using third-party geocoders and other data sources.

geopy is tested against CPython (versions 2.7, 3.4, 3.5, 3.6, 3.7, 3.8), PyPy, and PyPy3. geopy does not and will not support CPython 2.6.

Geocoders

Each geolocation service you might use, such as Google Maps, Bing Maps, or Nominatim, has its own class in `geopy.geocoders` abstracting the service's API. Geocoders each define at least a `geocode` method, for resolving a location from a string, and may define a `reverse` method, which resolves a pair of coordinates to an address. Each Geocoder accepts any credentials or settings needed to interact with its service, e.g., an API key or locale, during its initialization

We will be using Nomination api to extract lat long of cities as shown below

```
[74] import time
     geolocator = Nominatim(user_agent="IBM project")
     def lat_long_getter(city):
       time.sleep(1)
       location = geolocator.geocode(city)
       if location is not None:
         return (location.latitude, location.longitude)
```

Output DF from geolocator

```
[104] chennai_cities_df.head(5)
```

	cities	lat	long
0	Adyar	13.006450	80.257779
1	Adambakkam	12.982221	80.209121
2	Alapakkam	13.049901	80.165435
3	Alandur	13.002822	80.171919
4	Alwarpet	13.033860	80.254549

2.2.Cleaning Data and removing outliers

Removing null:

As there are only 10 cities which are having null in lat long I am dropping the data as it is less than 5 % of the data

```
[99] chennai_cities_df=pd.concat([chennai_suburbs_df,pd.DataFrame(lat_long_list,columns=['lat','long'])
```

```
[100] chennai_cities_df.shape
```

```
(201, 3)
```

```
[102] chennai_cities_df.dropna(how='any',inplace=True)
```

Code Text

```
[103] chennai_cities_df.shape
```

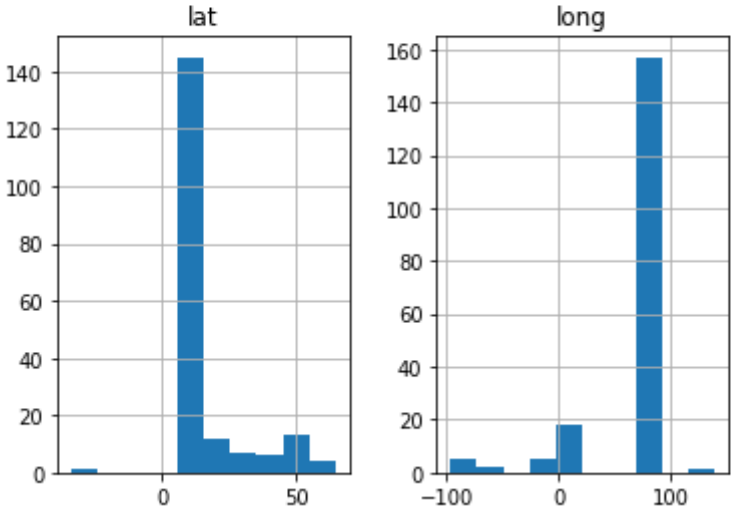
```
(188, 3)
```

Removing Outliers

We will be cleaning data and will be removing outliers by using histogram

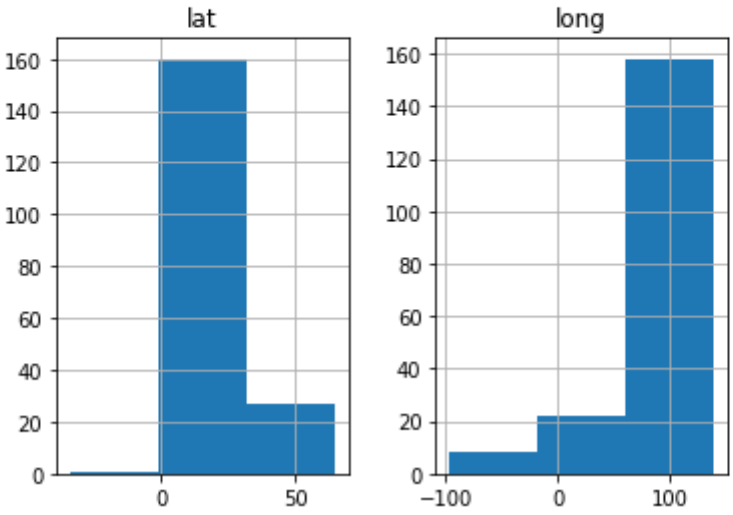
```
chennai_cities_df.hist()
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7f82ee0c
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f82edab
       dtype=object)
```




```
[108] chennai_cities_df.hist(bins=3)
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7
        <matplotlib.axes._subplots.AxesSubplot object at 0x7
       dtype=object)
```

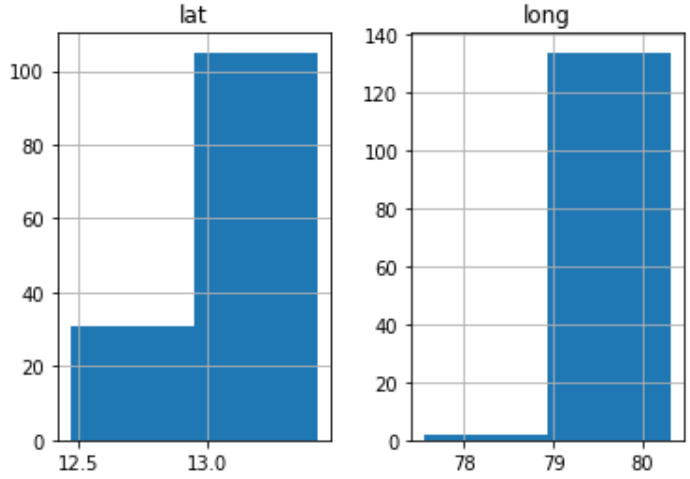


On seeing the data we can see the lat is centred around arount 10 and long is centred around 80

So I will be removing the outliers and we can see the distribution after removing outlier

```
[122] outlier_removed_df.hist(bins=2
                              )
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7f82ee106320>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f82ec443dd8>]]
      dtype=object)
```



Step3: Use foursquare api to find different venue

Next we will foursquare api and get diff venues around each city.
Since I am focussing on constructing apartment I will take only the places which have restaurant and Transport nearby so I will be filtering only records which have restaurant and stations nearby

```
[142] venues_df_filter=venues_df[venues_df['type'].str.contains('Station|Restaurant|Market')]
```

```
[145] venues_df_filter.shape ,venues_df.shape
```

```
((1580, 7), (3839, 7))
```

```
[150] # one hot encoding
      venues_df_filter_onehot = pd.get_dummies(venues_df_filter[['type']], prefix="", prefix_sep="")

      venues_df_one_hot_df=pd.concat([venues_df_filter[:-1],venues_df_filter_onehot],axis=1)
```

```
[153] venues_df_group= venues_df_one_hot_df.groupby('city').mean().reset_index()
```

Step4: Cluster places based on similarity using K-Means

We will be clustering using K-Means clustering and visualize the cluster

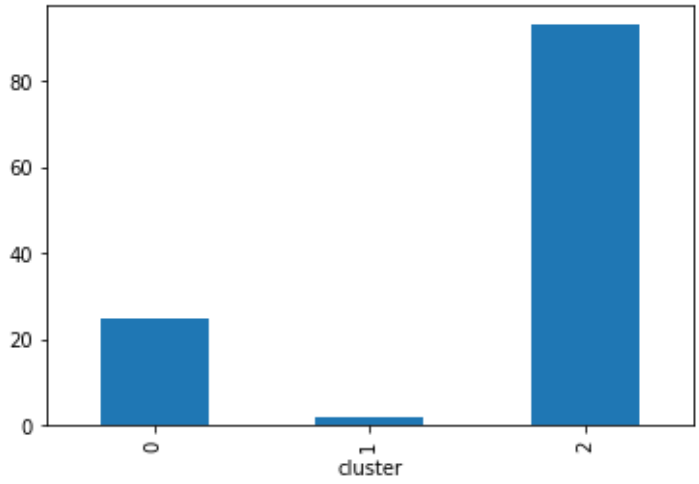
```
# set number of clusters
kclusters = 3

kl_clustering = venues_df_group.drop(["city"], 1)

# run k-means clustering
kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(kl_clustering)

# check cluster labels generated for each row in the dataframe
kmeans.labels_[0:10]
```

```
array([2, 2, 2, 2, 2, 2, 2, 2, 2, 2], dtype=int32)
```

Visualizing Cluster

```
[168] venues_df_group.groupby('cluster').count()['city'].plot(kind='bar')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f82ec36a518>
```



Step5 :Conclusion

On seeing the above cluser we find cluster 1 is small and We can choose the small cluster so that we can extract the above region as on inspecting the records the area is in main area and is nearby bus stations as well

```
[171] venues_df_group[venues_df_group['cluster']==1]
```

	city	lat	lon	loc_lat	loc_long	Afghan Restaurant	African Restaurant	American Restaurant	Andhra Restaurant	R
86	Shenoy	12.930837	77.584122	12.926998	77.583580	0.0	0.0	0.0	0.022222	
103	Tolgate	12.978422	77.548794	12.980337	77.544139	0.0	0.0	0.0	0.000000	