Department of Computer Science and Engineering
College of Engineering, Guindy
Anna University, Chennai – 600 025

Exploratory Data Analysis of Geolocational Data for Student Accommodation

NAME	REG.NO	E-MAIL	MOBILE NO
Gokulnath M	2019103522	gokulmanickam99@gmail.com	7708578759
Kavishree S	2019103537	kavi10092000@gmail.com	8300687349

INSTRUCTOR: **Dr AROCKIA XAVIER ANNIE R**

Contents

1.ABSTRACT:

In the fast moving, effort-intense environment that a student inhabits, It's a frequent occurrence that one is too tired to fix oneself a home-cooked meal. And of course, even if one gets home-cooked meals every day, it is not unusual to want to go out for a good meal every once in a while for social/recreational purposes. Either way, it's a commonly understood idea that regardless of where one lives, the food one eats is an important aspect of the lifestyle one leads. Now, imagine a scenario where a person has newly moved into a new location. They already have certain preferences, certain tastes. It would save both the student and the food providers a lot of hassle if the student lived close to their preferred outlets. Convenience means better sales, and saved time for the customer. K-means clustering is an ideal algorithm to identify unknown trends, so it is best suited to our problem. We use four square API to get geolocational data and identify the top categories that represented the Food and Grocery categories, as well as restaurants which were amusingly overshadowed by bus stops and apply K-Means. Based on the formation of cluster the students can choose where they accommodate.

2.INTRODUCTION:

This project uses K-Means Clustering to find best accommodation of students around their college by classifying accommodation for incoming students on the basis of their preferences on amenities, budget and proximity to the location. Food delivery apps aside, managers of restaurant chains and hotels can also leverage this information. For example, if a manager of a restaurant already knows the demographic of his current customers, they'd ideally want to open at a location where this demographic is at its highest concentration, ensuring short commute times to the location and more customers served. If potential hotel locations are being evaluated, a site that caters to a wide variety of tastes would be ideal, since one would want every guest to have something to their liking.

3.LITERATURE SURVEY:

[1] A Geo-data Collection Strategy to Assess Housing in Its Social, Environmental, and Spatial Aspects

Blocks of flats are the most abundant built element in many cities and their quality is an important issue. Undergraduate students often occupy these accommodations and this group describes a wide range of housing situations that makes it possible to assess their quality. This research aims to describe the data collection strategy in an ongoing research project in the city of A Coruna (Spain), oriented towards assessing the ordinary blocks of flats where the students live (shared flats, apartments). The method uses several sources (a questionnaire, cartographic viewers, and municipal historical archives) and a sophisticated strategy to be non-intrusive, efficient, and user-friendly. This information would allow an understanding of both students' changes of residence and their locations within the city. Formal methods are applied to estimate ordinary interior designs based on external configurations and consequently verify them with users and public archives. This holistic strategy creates a georeferenced database by adopting efficient procedures, thus reducing user inquiries. The case of A Coruna confirms the variability of housing conditions and the complex phenomena involved, including possible correlations between housing quality and economic/social parameters, which can be studied with the developed database.

[2] Using Foursquare place data for estimating building block use

Information about the land use of built-up areas is required for the comprehensive planning and management of cities. However, due to the high cost of the land use surveys, land use data is outdated or not available for many cities. Therefore, we propose the reuse of up-to-date and low-cost place data from social media applications for land use mapping purposes. As main case study, we used Foursquare place data for estimating nonresidential building block use in the city of Amsterdam. Based on the Foursquare place categories, we estimated the use of 9827 building blocks, and we compared the classification results with a reference building block use dataset. Our evaluation metric is the kappa coefficient, which determines if the classification results are significantly better than a random guess result. Using the optimal set of parameter

values, we achieved the highest kappa coefficient values for the land use categories “hotels, restaurants and cafes” (0.76) and “retail” (0.65). The lowest kappa coefficients were found for the land use categories “industries” and “storage and unclear”. We have also applied the methodology in another case study area, the city of Varese in Italy, where we had similar accuracy results. We therefore conclude that Foursquare place data can be trusted only for the estimation of particular land use categories.

[3]Process Discovery on geolocational data

Fleet tracking technology collects real-time information about geolocation of vehicles as well as driving-related data. This information is typically used for location monitoring as well as for analysis of routes, vehicles and drivers. From an operational point of view, the geolocation simply identifies the state of a vehicle in terms of positioning and navigation. From a management point of view, the geolocation may be used to infer the state of a vehicle in terms of process (e.g., driving, fueling, maintenance, or lunch break). Meaningful information may be extracted from these inferred states using process mining. An innovative methodology for inferring process states from geolocation data is proposed in this paper. Also, it is presented the potential of applying process mining techniques on geolocation data for process discovery.

4.PROBLEM STATEMENT:

Imagine a scenario where a person has newly moved into a new location. They already have certain preferences, certain tastes. It would save both the student and the food providers a lot of hassle if the student lived close to their preferred outlets. Convenience means better sales, and saved time for the customer.

5.PROBLEM SOLUTION:

This project involves the use of K-Means Clustering to find the best accommodation for students in Bangalore (or any other city of your choice) by classifying accommodation for incoming students on the basis of their preferences on amenities, budget and proximity to the location.

5.1 DATASET:

Dataset includes information on food choices, nutrition, preferences, childhood favourites, and other information from college students. There are 126 responses from students. Data is raw and uncleaned. There are around 70 parameters.

General idea of the dataset:

employment - do you work?

1 - yes full time
2 - yes part time
3- no
4 - other

eating_out - frequency of eating out in a typical week

1 – Never
2 - 1-2 times
3 - 2-3 times
4 - 3-5 times
5 - every day

income

1 - less than $15,000

2 - $15,001 to $30,000

3 - $30,001 to $50,000

4 - $50,001 to $70,000

5 - $70,001 to $100,000

6 - higher than $100,000

cook - how often do you cook?

1 - Every day

2 - A couple of times a week

3- Whenever I can, but that is not very often

4 - I only help a little during holidays

5 - Never, I really do not know my way around a kitchen

how often do you exercise in a regular week?

1 – Everyday

2 - Twice or three times per week

3 - Once a week

4 - Sometimes

5 - Never

5.2 INPUT:

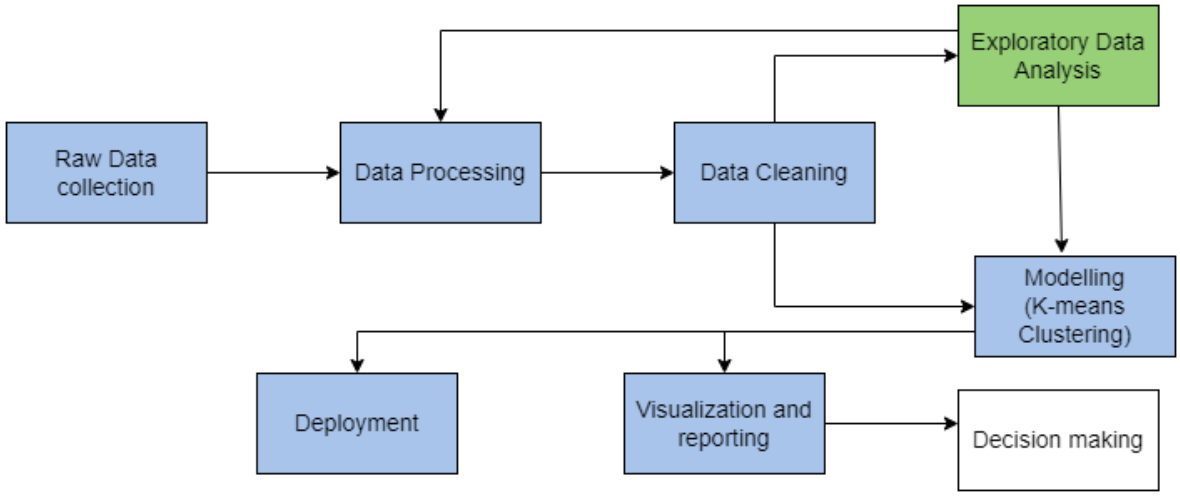
Fetch Geolocational Data from the Foursquare API.

5.3 APPROACH:

K Means Clustering will help us group locations based on the amenities located around them. For example, a location with a high amount of shops nearby will be labelled "Amenity Rich" while a location with less amenities will be labelled "Amenity Poor". Similar locations will be grouped (clustered) together.

6.ARCHITECTURE :

6.1 ARCHITECTURE DIAGRAM:



6.2 EXPLANATION OF THE ARCHITECTURE:

Fetch Datasets from the relevant locations (Data Collection). Clean the Datasets to prepare them for analysis. (Data Cleaning via Pandas). Visualise the data using boxplots. (Using Matplotlib /Seaborn /Pandas). Fetch Geolocational Data from the Foursquare API. (REST APIs). Use K-Means Clustering to cluster the locations (Using ScikitLearn). Present findings on a map. (Using Folium/Seaborn).

8.LIST OF MODULES:

8.1 Data Preprocessing

8.2 Data Exploration and Visualization

8.3 K-Means Clustering on the data

8.4 Plot the Clustered locations on a map

8.1 DATA PREPROCESSING

To clean the Foursquare data to a usable state, drop all irrelevant entries like location.cc, country, city, hasPerk ,state etc. and retained only information like Location Name, Address, and its latitude and longitude. The problem with Foursquare data is that it is very finicky to call the API: sometimes the queries would return 2, sometimes 50 locations depending on the search word. The results varied wildly with minute changes. Similarly for food_coded dataset irrelevant entries are dropped.

Food coded dataset

```
[ ] dfstudents.head()
```

	GPA	Gender	breakfast	calories_chicken	calories_day	calories_scone	coffee	comfort_food	comfort_food_reasons	comfort_food_reasons_coded	...
0	2.4	2	1	430	NaN	315.0	1	none	we dont have comfort	9.0	...
1	3.654	1	1	610	3.0	420.0	2	chocolate, chips, ice cream	Stress, bored, anger	1.0	...
2	3.3	1	1	720	4.0	420.0	2	frozen yogurt, pizza, fast food	stress, sadness	1.0	...
3	3.2	1	1	430	3.0	420.0	2	Pizza, Mac and cheese, ice cream	Boredom	2.0	...
4	3.5	1	1	720	2.0	420.0	2	Ice cream, chocolate, chips	Stress, boredom, cravings	1.0	...

5 rows × 61 columns

After dropping irrelevant entries.

```
[ ] dfclean=dfstudents[['cook','eating_out','employment','ethnic_food','exercise','fruit_day','income','on_off_campus','pay_meal_out','sports','veggies_day']]
    dfclean.dropna(axis=0,inplace=True)
    dfclean.head()
```

```
/usr/local/lib/python3.7/dist-packages/pandas/util/_decorators.py:311: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  return func(*args, **kwargs)
```

	cook	eating_out	employment	ethnic_food	exercise	fruit_day	income	on_off_campus	pay_meal_out	sports	veggies_day
0	2.0	3	3.0	1	1.0	5	5.0	1.0	2	1.0	5
1	3.0	2	2.0	4	1.0	4	4.0	1.0	4	1.0	4
2	1.0	2	3.0	5	2.0	5	6.0	2.0	3	2.0	5
3	2.0	2	3.0	5	3.0	4	6.0	1.0	2	2.0	3
4	1.0	2	2.0	4	1.0	4	6.0	1.0	4	1.0	4

Get Geolocational Data from Foursquare API

Latitude and Longitude of Anna university college of engineering, Guindy location

```
search_query = 'Apartment' #Search for residential locations
radius = 18000 #Set the radius to 18 kilometres due to traffic constraints
latitude=13.011624#College location
longitude=80.234697
```

```
CLIENT_ID = '5O2OQN1HOFBER5CQ4VJSEBU3PZ54Z3T31EGTNFJ0V04KSKZI'
CLIENT_SECRET = 'H2KIZ3QZRGZD2GVCOSSTL04SQ3N0Y5QWEWQKPQ1VTPQNF24R'
VERSION = '20180604'
LIMIT = 500
```

```
url = 'https://api.foursquare.com/v2/venues/search?client_id={}&client_secret={}&ll={},{}&v={}&query={}&radius={}&limit={}'.format(CLIENT_ID, CLIENT_SECRET, latitude, longitude, VERSION, search_query, radius, LIMIT)
```

```
results = requests.get(url).json()
```

```
# assign relevant part of JSON to venues
venues = results['response']['venues']

# tranform venues into a dataframe
dataframe = json_normalize(venues)
dataframe.head()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5: FutureWarning: pandas.io.json.json_normalize is deprecated, use pandas.
  """
```

	id	name	categories	referralId	hasPerk	location.address	location.lat	location.lng	l
0	51bb4731498e79e3b04c3d59	Apartment	[{'id': '4bf58dd8d48988d1c4941735', 'name': 'R...	v-1654419021	False	The Park Hyatt	13.009774	80.220027	
1	4dfd74ab227185f38b98a592	Ficus Grove Apartment	[{'id': '4d954b06a243a5684965b473', 'name': 'R...	v-1654419021	False	72, M.R.C Nagar Main Rad	13.019830	80.272997	
2	4defacdb52b1d9bad3c500d2	NFL's Apartment	[{'id': '4d954b06a243a5684965b473', 'name': 'R...	v-1654419021	False	4th Main Rd, Kasturba Nagar, Adyar	13.004583	80.251305	
3	54ea9c60498e2c090a1cea5c	Senna Apartment	[{'id': '4d954b06a243a5684965b473', 'name': 'R...	v-1654419021	False	NaN	13.021061	80.217049	
4	5798f6db498e83026abbd672	Euphoria Apartment	[{'id': '4d954b06a243a5684965b473',	v-1654419021	False	NaN	12.936379	80.202414	

After dropping irrelevant entries.

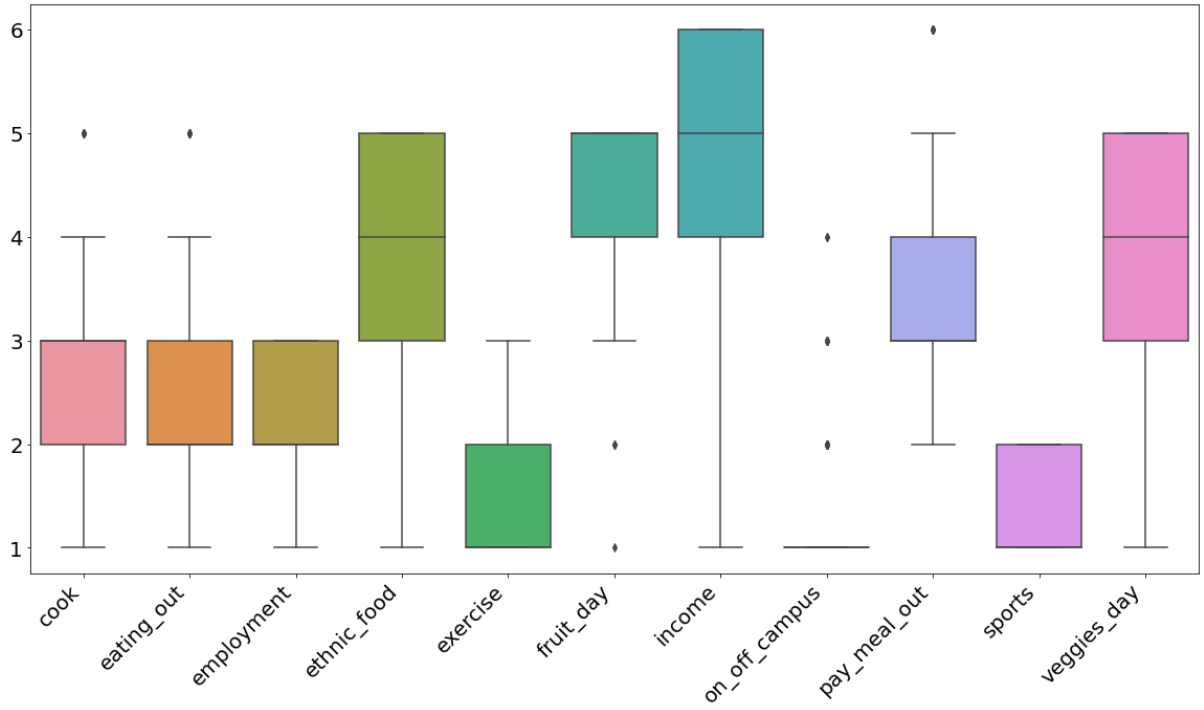
```
filtered_columns = ['name', 'categories'] + [col for col in dataframe.columns if col.startswith('location.')] + ['id']
dataframe_filtered = dataframe.loc[:, filtered_columns]

# 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']

# filter the category for each row
dataframe_filtered['categories'] = dataframe_filtered.apply(get_category_type, axis=1)

# clean column names by keeping only last term
dataframe_filtered.columns = [column.split('.')[-1] for column in dataframe_filtered.columns]
dataframe_filtered.drop(['cc','country','state','city'],axis=1,inplace=True)
dataframe_filtered.head()
```

	name	categories	address	lat	lng	labeledLatLngs	distance	formattedAddress	crossStreet	postalCode	id
0	Apartment	Restaurant	The Park Hyatt	13.009774	80.220027	[{'label': 'display', 'lat': 13.00977410383588...	1604	[The Park Hyatt, India]	NaN	NaN	51bb4731498e79e3b04c3d59
1	Ficus Grove Apartment	Residential Building (Apartment / Condo)	72, M.R.C Nagar Main Rad	13.019830	80.272997	[{'label': 'display', 'lat': 13.01983021086314...	4253	[72, M.R.C Nagar Main Rad (R.A. Puram), Chenna...	R.A. Puram	600 028	4dfd74ab227185f38b98a592
2	NFL's Apartment	Residential Building (Apartment / Condo)	4th Main Rd, Kasturba Nagar, Adyar	13.004583	80.251305	[{'label': 'display', 'lat': 13.0045832, 'lng':...	1964	[4th Main Rd, Kasturba Nagar, Adyar, Chennai, ...	NaN	NaN	4defacdb52b1d9bad3c500d2
3	Senna Apartment	Residential Building (Apartment / Condo)	NaN	13.021061	80.217049	[{'label': 'display', 'lat': 13.02106094360351...	2183	[India]	NaN	NaN	54ea9c60498e2c090a1cea5c
4	Euphoria Apartment	Residential Building (Apartment / Condo)	NaN	12.936379	80.202414	[{'label': 'display', 'lat': 12.936379, 'lng':...	9078	[3rd Cross street, Chennai 600100, Tamil Nadu,...	3rd Cross street	600100	5798f6db498e83026abbd672

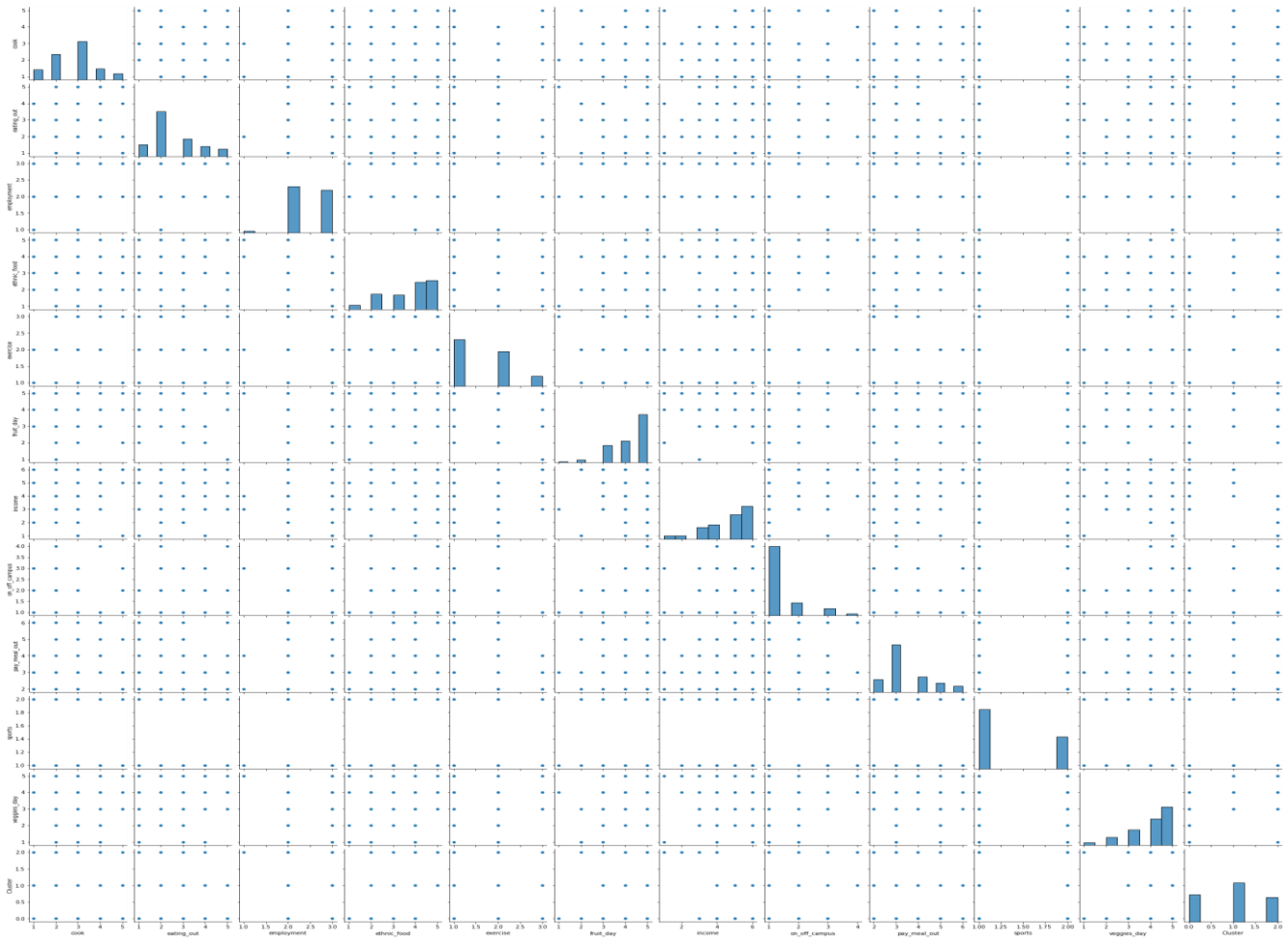
8.2 DATA EXPLORATION AND VISUALIZATION

Boxplots are a measure of how well distributed the data in a data set is. It divides the data set into three quartiles. This graph represents the minimum, maximum, median, first quartile and third quartile in the data set. It is also useful in comparing the distribution of data across data sets by drawing boxplots for each of them. Here x-axis denotes the data to be plotted while the y-axis shows the frequency distribution.

```
plt.figure(figsize=(20, 10))
#plt.xticks(rotation='vertical')
sns.boxplot

ax = sns.boxplot(data = dfclean)
ax.tick_params(labelsize=20)
plt.xticks(rotation=45, ha='right')

plt.show()
```



Pairplot visualizes the given data to find the relationship between them.



8.3 K-MEANS CLUSTERING ON THE DATA

Using elbow method to find optimized K value, and visualizing the clusters in boxplot.

ALOGRITHM

The working of the K-Means algorithm is explained in the below steps:

Step-1: Select the number K to decide the number of clusters.

Step-2: Select random K points or centroids.

Step-3: Assign each data point to their closest centroid, which will form the predefined K clusters.

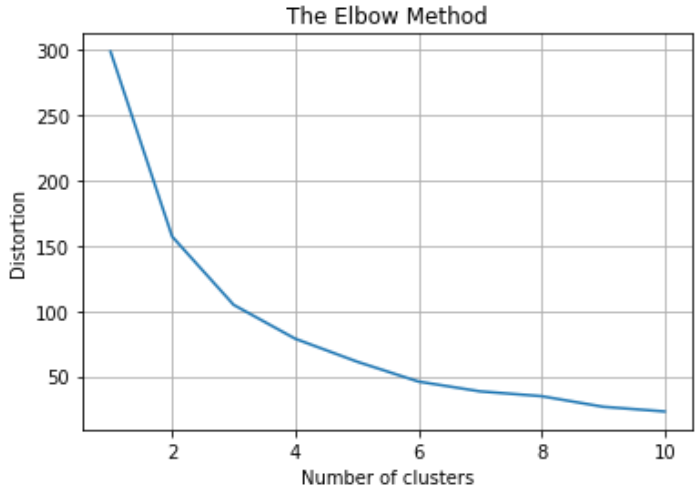
Step-4: Calculate the variance and place a new centroid of each cluster.

Step-5: Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.

Step-6: If any reassignment occurs, then go to step-4 else go to FINISH.

Step-7: The model is ready.

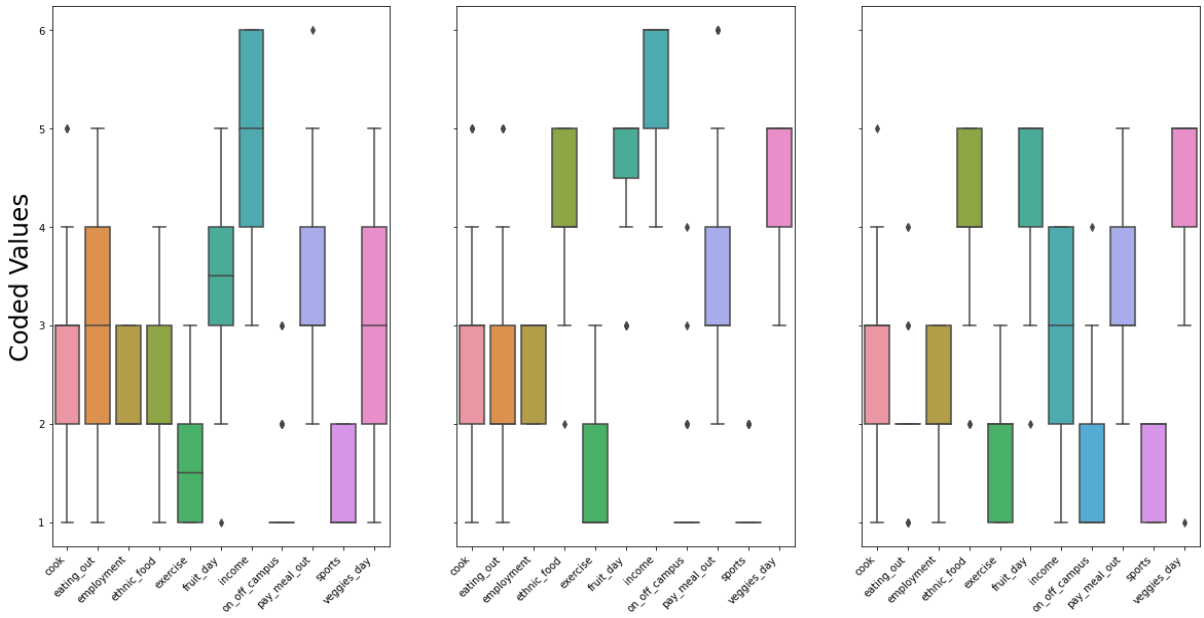
```
f=['cook','income']
X = dfclean[f]
max_k = 10
## iterations
distortions = []
for i in range(1, max_k+1):
    if len(X) >= i:
       model = KMeans(n_clusters=i, init='k-means++', max_iter=300, n_init=10, random_state=0)
       model.fit(X)
       distortions.append(model.inertia_)
## best k: the lowest derivative
k = [i*100 for i in np.diff(distortions,2)].index(min([i*100 for i in np.diff(distortions,2)]))
## plot
fig, ax = plt.subplots()
ax.plot(range(1, len(distortions)+1), distortions)
ax.set(title='The Elbow Method', xlabel='Number of clusters',
       ylabel="Distortion")
ax.grid(True)
plt.show()
```



From elbow method ,setting clusters=3

```
kclusters = 3

# run k-means clustering
kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(dfclean)
dfclean['Cluster']=kmeans.labels_
```

```
fig, axes = plt.subplots(1,kclusters, figsize=(20, 10), sharey=True)
axes[0].set_ylabel('Coded Values', fontsize=25)

for k in range(kclusters):
     plt.sca(axes[k])
     plt.xticks(rotation=45,ha='right')
     sns.boxplot(data = dfclean[dfclean['Cluster'] == k].drop('Cluster',1), ax=axes[k])

plt.show()
```



3 clusters provide three clear groups: One high income group which insists on eating healthy, the other is more relaxed in diet.

The low income group has its own profile, as discussed in insights.

For higher numbers of clusters, it becomes difficult to draw any concrete conclusions.

High income students in general are:

1.>Much more likely to stay on campus.

2.>Less likely to cook.

3.>Eat out more often.

In addition, a subset of higher income students: 1.>Eat out less often.

2.>Have a healthier, more varied diet in general. (Eat ethnic food, fruits, vegetables etc.)

Lower income students are more likely to:

1.>Eat more vegetables

2.>Eat more fruits.

3.>More likely to cook, so more likely to stay off campus.

4.>Obviously, pay less for meals out.

8.4 PLOT THE CLUSTERED LOCATIONS ON A MAP

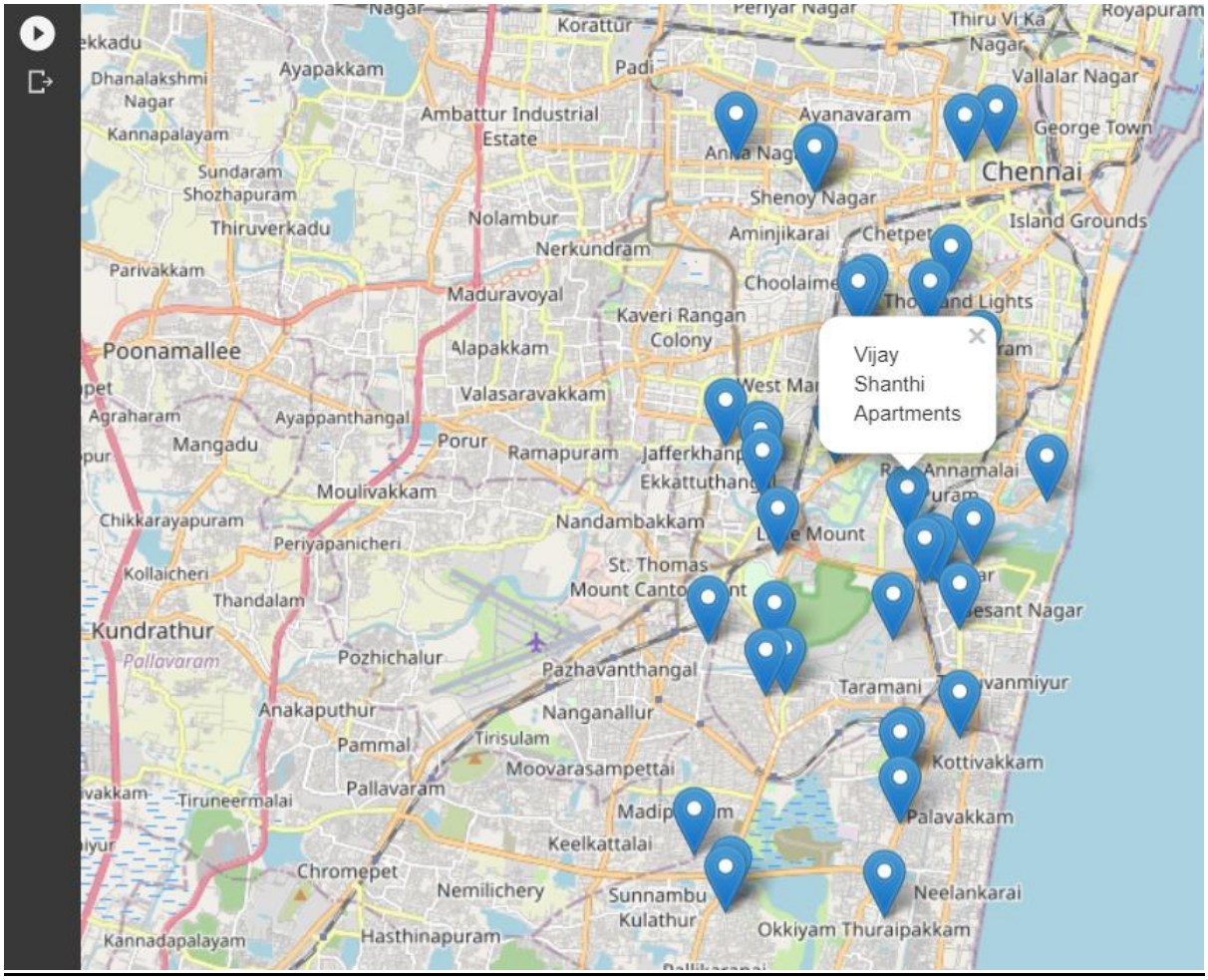
Using KMeans clustering to plot locations on map. Retrieving restaurants, fruits, groceries etc near the apartments to cluster.

Plot apartments on the map.

```
map_bang=folium.Map(location=[13.011624,80.234697],zoom_start=12)

locations = folium.map.FeatureGroup()

latitudes = list(dataframe_filtered.lat)
longitudes = list( dataframe_filtered.lng)
labels = list(dataframe_filtered.name)

for lat, lng, label in zip(latitudes, longitudes, labels):
    folium.Marker([lat, lng], popup=label).add_to(map_bang)

# add incidents to map
map_bang.add_child(locations)

# add incidents to map
map_bang.add_child(locations)

map_bang
```

```
[ ] df_evaluate=dataframe_filtered[['lat','lng']]
```

```
[ ] RestList=[]
    latitudes = list(dataframe_filtered.lat)
    longitudes = list( dataframe_filtered.lng)
    for lat, lng in zip(latitudes, longitudes):
        radius = 5000 #Set the radius to 5 kilometres for convenience
        latitude=lat#Query for the apartment location in question
        longitude=lng
        url = 'https://api.foursquare.com/v2/venues/search?client_id={}&client_secret={}&ll={},{}&v={}&query={}&radius={}&limit={}'.format(CLIENT_ID, CLIENT_SECRET, latitude, longitude, VERSION, search
        search_query = 'Restaurant' #Search for any food related locations
        results = requests.get(url).json()
        # assign relevant part of JSON to venues
        venues = results['response']['venues']
        # tranform venues into a dataframe
        dataframe2 = json_normalize(venues)
        filtered_columns = ['name', 'categories'] + [col for col in dataframe2.columns if col.startswith('location.')] + ['id']
        dataframe_filtered2 = dataframe2.loc[:, filtered_columns]
        # filter the category for each row
        dataframe_filtered2['categories'] = dataframe_filtered2.apply(get_category_type, axis=1)
        # clean column names by keeping only last term
        dataframe_filtered2.columns = [column.split('.')[-1] for column in dataframe_filtered2.columns]
        RestList.append(dataframe_filtered2['categories'].count())
```




```
[ ]  df_evaluate['Restaurants']=RestList
```

```
FruitList=[]
latitudes = list(dataframe_filtered.lat)
longitudes = list( dataframe_filtered.lng)
for lat, lng in zip(latitudes, longitudes):
    radius = 5000 #Set the radius to 5 kilometres for convenience
    latitude=lat#Query for the apartment location in question
    longitude=lng
    url = 'https://api.foursquare.com/v2/venues/search?client_id={}&client_secret={}&ll={},{}&v={}&query={}&radius={}&limit={}'.format(CLIENT_ID, CLIENT_SECRET, latitude, l
    search_query = 'Fruit' #Search for any food related locations
    results = requests.get(url).json()
    # assign relevant part of JSON to venues
    venues = results['response']['venues']
    # tranform venues into a dataframe
    dataframe2 = json_normalize(venues)
    filtered_columns = ['name', 'categories'] + [col for col in dataframe2.columns if col.startswith('location.')] + ['id']
    dataframe_filtered2 = dataframe2.loc[:, filtered_columns]
    # filter the category for each row
    dataframe_filtered2['categories'] = dataframe_filtered2.apply(get_category_type, axis=1)
    # clean column names by keeping only last term
    dataframe_filtered2.columns = [column.split('.')[-1] for column in dataframe_filtered2.columns]
    FruitList.append(dataframe_filtered2['categories'].count())
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:14: FutureWarning: pandas.io.json.json_normalize is deprecated, use pandas.json_normalize instead
```

```
[ ]  df_evaluate['Fruits,Vegetables,Groceries']=FruitList
```

```
kclusters = 3

# run k-means clustering
kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(df_evaluate)
df_evaluate['Cluster']=kmeans.labels_
df_evaluate['Cluster']=df_evaluate['Cluster'].apply(str)
df_evaluate.head(10)
```

	lat	lng	Restaurants	Fruits,Vegetables,Groceries	Cluster
0	13.009774	80.220027	47	47	0
1	13.019830	80.272997	46	40	2
2	13.004583	80.251305	46	42	2
3	13.021061	80.217049	49	49	0
5	13.035335	80.236748	48	50	0
6	12.995446	80.255680	47	43	2
7	13.030906	80.258908	47	49	0
8	12.983294	80.221405	46	40	2
9	13.030500	80.209830	48	50	0
10	12.992836	80.206542	46	32	2

```
#define coordinates of the college 13.011624
map_bang=folium.Map(location=[13.011624,80.234697],zoom_start=12)
# instantiate a feature group for the incidents in the dataframe
locations = folium.map.FeatureGroup()
# set color scheme for the clusters
def color_producer(cluster):
    if cluster=='0':
        return 'green'
    elif cluster=='1':
        return 'orange'
    else:
        return 'red'
latitudes = list(df_evaluate.lat)
longitudes = list(df_evaluate.lng)
labels = list(df_evaluate.Cluster)
names=list(dataframe_filtered.name)
for lat, lng, label,names in zip(latitudes, longitudes, labels,names):
    folium.CircleMarker(
            [lat,lng],
            fill=True,
            fill_opacity=1,
            popup=folium.Popup(names, max_width = 300),
            radius=5,
            color=color_producer(label)
        ).add_to(map_bang)

# add locations to map
map_bang.add_child(locations)

map_bang
```

9.FORMULA:

$$\text{WCSS} = \sum_{C_k}^{C_n} \left(\sum_{d_i in C_i}^{d_m} distance(d_i, C_k)^2 \right)$$

Where,
C is the cluster centroids and d is the data point in each Cluster.

10.RESULTS:

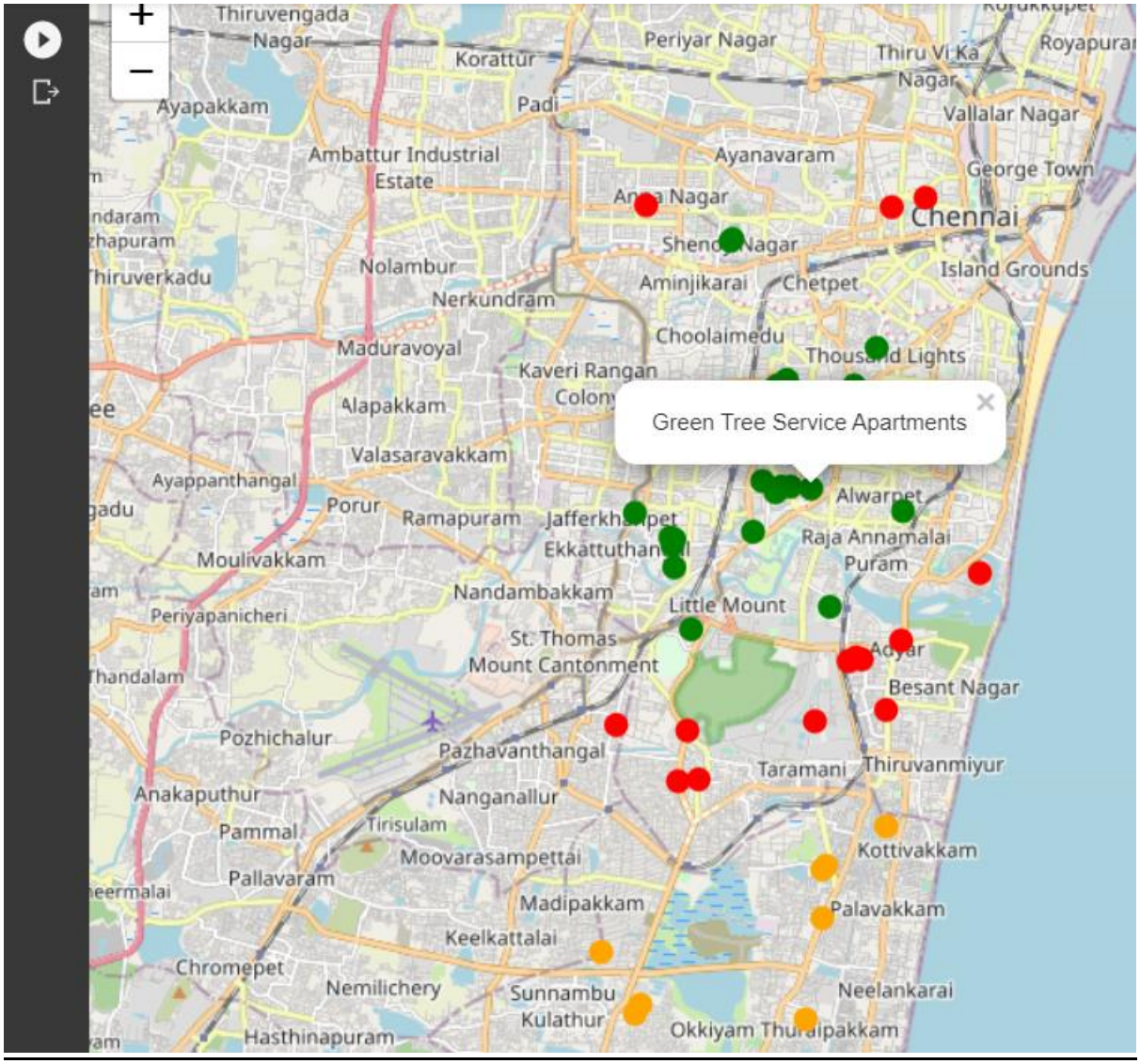
K-means Clustering on Student Data

- One cluster of high income students seem to eat out more often and spend more per meal, and care less about fruits, vegetables and ethnic food, and stay almost exclusively on campus.
- The second cluster of high income students seem to eat out less, pay less per meal, and are more likely to stay off campus.
- The cluster of low income students eat out less, and cook more often than the high income group. They eat as much vegetables as the second cluster but eat less fruits (perhaps because fruits are more expensive than vegetables?) and are most likely to stay off campus.

K-means Clustering on Location Data

Three prominent clusters emerged after applying the method on the data:

- Cluster O(Green) Where both (fruits and vegetables) and (restaurants) are abundant
- Cluster 1(Yellow): Restaurants are plentiful, but groceries less so.
- Cluster 2(Red): Restaurants and groceries are relatively hard to find.



11.CONCLUSION

K-means clustering can help find patterns where none are apparent. Income is a very useful factor to classify students behavior and spending patterns. Diet is another important factor to account for when looking at accommodation for students. Foursquare data is limited but can provide insights into a city's infrastructure. This data could be supplemented with other sources to provide better results.

12.References

1. Pereira-Martínez, D., López-Chao, V., Lizancos, P., & Borges Pereira, V. (2022). A Geo-data Collection Strategy to Assess Housing in Its Social, Environmental, and Spatial Aspects.
2. Using Foursquare place data for estimating building block use
   Spyridon Spyratos, Demetris Stathakis, Michael Lutz, , , Chrisa Tsinaraki.
3. Joel Riberio, Tania Fontes, Carlos Soares, Joseluis Borges. Process Discovery on geolocational data, September 2019.

13.Appendix

The undersigned acknowledge they have completed implementing the project “Flight Delay Prediction” and agree with the approach it presents.

Signature: _____ Date: 06/06/2021

Name: Gokulnath M

Signature: _____ Date: 06/06/2021

Name: Kavishree S