In [*]:
```
import pandas as pd
from matplotlib import pyplot as plt
from scipy import spatial
import numpy as np
from sklearn.cluster import KMeans
```

Read and clear data. We assumed that all rows in this table belong to clients (to gain time,as a #of clients >> #of sellers)

In [133]:
```
df = pd.read_csv('.\data\olist_geolocation_dataset.csv')
```

In [129]:
```
df.shape
```

Out[129]:
```
(1000121, 5)
```

In [10]:
```
# Removing some outliers
#Brazils most Northern spot is at 5 deg 16' 27.8" N latitude.;
df = df[df.geolocation_lat <= 5.27438888]
#it’s most Western spot is at 73 deg, 58' 58.19"W Long.
df = df[df.geolocation_lng >= -73.98283055]
#It’s most southern spot is at 33 deg, 45' 04.21" S Latitude.
df = df[df.geolocation_lat >= -33.75116944]
#It’s most Eastern spot is 34 deg, 47' 35.33" W Long.
df = df[df.geolocation_lng <=  -34.79314722]
```

In [11]:
```
df['geolocation_city'].value_counts()
```

Out[11]:
```
sao paulo                      135800
rio de janeiro                  62151
belo horizonte                  27805
são paulo                       24918
curitiba                        16593
porto alegre                    13521
salvador                        11865
guarulhos                       11340
brasilia                        10470
sao bernardo do campo            8112
osasco                           7658
santo andre                      6863
niteroi                          6534
recife                           6168
goiania                          5661
fortaleza                        5538
campinas                         5479
sorocaba                         5361
santos                           5000
barueri                          4971
juiz de fora                     4679
contagem                         4395
campo grande                     4332
ribeirao preto                   4187
florianopolis                    4148
nova iguacu                      4022
mogi das cruzes                  3913
belem                            3789
sao jose dos campos              3759
sao goncalo                      3601
                                ...
cuiabá paulista                     1
araçaí                              1
caldeirão grande                    1
socorro do piaui                    1
santa elvira                        1
iauarete                            1
flor do sertão                      1
joselândia                          1
serra da tapuia                     1
anama                               1
ana dias                            1
pinheiros altos                     1
petunia                             1
boa sorte                           1
são josé das missões                1
ipecaetá                            1
isaías coelho                       1
santo antonio do manhuacu           1
potunduva                           1
eleuterio                           1
santanopolis                        1
vila dos cabanos                    1
pedro avelino                       1
maraa                               1
jacobina do piaui                   1
união de minas                      1
matrinchã                           1
josé gonçalves de minas             1
fatimarmnte dutra                   1
ribeiro gonçalves                   1
Name: geolocation_city, Length: 8006, dtype: int64
```

The idea to make a segmentation by geographical position is following.

- We could not use segmentation by country, as we have all clients in Brazil.
- All clients live in more or less similar timezone. So we could not use this type of segmentations.

But we could divide geo data by region of poulation density. For marketing point of view it would be interesting to consider:

1. regions where we have high density of clients (big towns);
2. regions near the town where population density is lower than in the towns;
3. regions with low density (villages, suburbs).

   Because according these categories marketing department could build advertising campagnies. Because clients in different segments have differents interest. For example, there is less demand of garden equipments from clients who live in town.

In [ ]:
```
X=df[['geolocation_lat','geolocation_lng']]
Y=df[['geolocation_zip_code_prefix']]
df=df.iloc[:1000]
```

We precalculate distance between all clients by building KDtree

In [20]:
```
tree = spatial.KDTree(df[['geolocation_lat','geolocation_lng']])
```

Lets get distances for near 3 neighbors for each client

In [ ]:
```
distances, neighbors_idx = tree.query(X, k=3)
```

Calculate avg distance to the neighbors for each client

In [69]:
```
avg_dist = np.average(distances, axis=1)
```

Now we want to obtain 3 cluster based on avg distance between clients, to identify high, medium and low population density region in data set. For this purpose we could use Kmeans algo (it's not the best for 1dim problem, but for testing reason it's ok).

In [73]:
```
avg_dist=avg_dist.reshape(-1,1)
```

In [74]:
```
kmeans=KMeans(n_clusters=3).fit(avg_dist)
```

Lets build datasets according to our clusters.

In [ ]:
```
first_segment = df.iloc[np.where(kmeans.labels_ ==0)]
second_segment = df.iloc[np.where(kmeans.labels_ ==1)]
third_segment = df.iloc[np.where(kmeans.labels_ ==2)]
```

In [ ]:
```

```