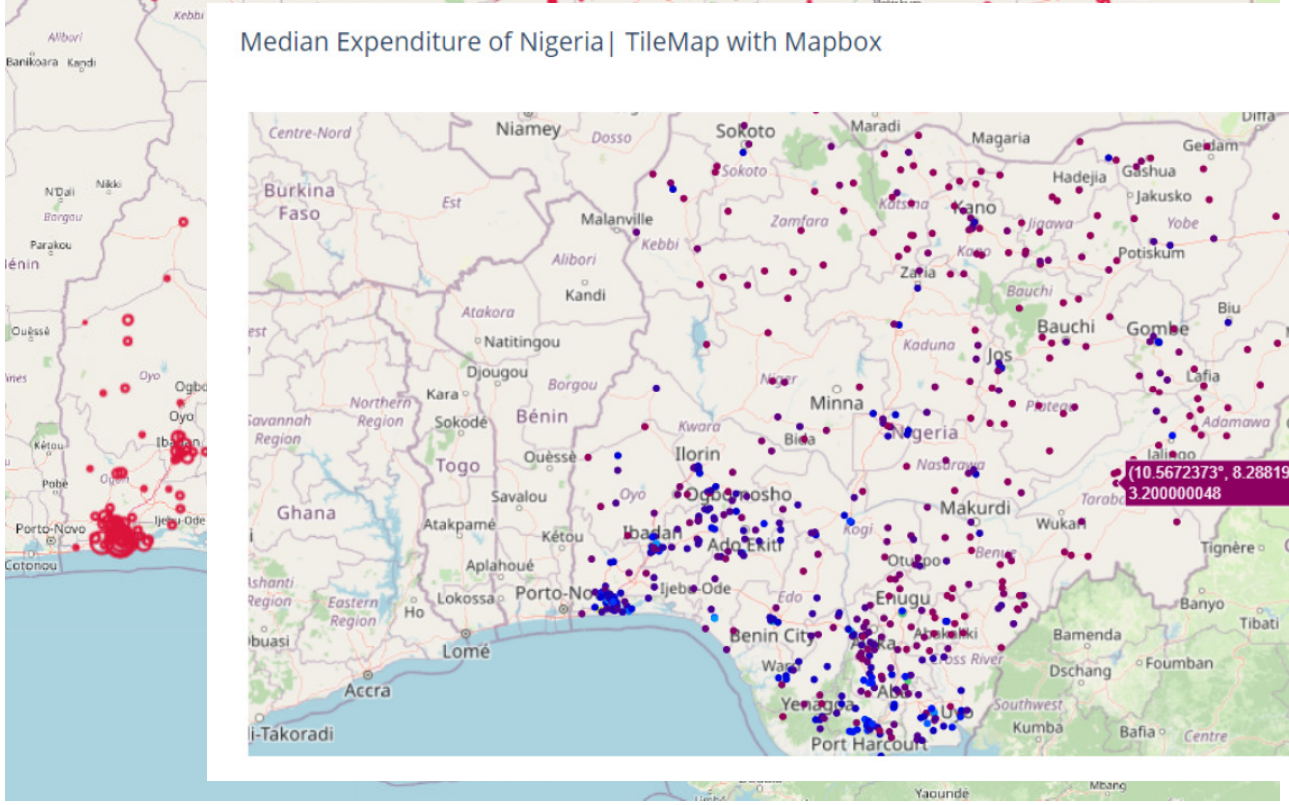
Nigerian Neighborhoods Median Expenditure Predictions

Sreya Dhar
Data Science MS'22
University at Buffalo

Locations vs. median expenditures of Nigerian neighborhoods



Part-1 :: Case Study on Nigerian Neighborhood to predict Median expenditure

Algorithms		Hyper parameters	Accuracy Matrices in %	
			Train Set	Test Set
Polynomial Regression		--	48.9	42.16
Random Forest		500 trees	83.12	54.01
XGBoost		learning_rate= 0.01, max_depth= 2, n_estimators= 150	75.32	56.92
Decision Tree		max_depth = 7, n_estimators = 1	75.68	46.41
KNN		k=3	60.34	47.50
AdaBoost		max_depth = 7, n_estimators = 500	53.36	79.48
S V M	Linear	--	34.95	34.66
	Radial	--	34.91	34.62
	Poly.	Degree = 14	63.48	50.49

Spatial regression (model-1)

```
from pysal.model import spreg
# Fit OLS model
m1 = spreg.OLS(
    y = data_norm[['median_spend']].values,
    x = data_norm[data_norm.columns[3:].tolist()].values,
    name_y='median_spend',
    name_x=data_norm.columns[3:].tolist())

print(m1.summary)
```

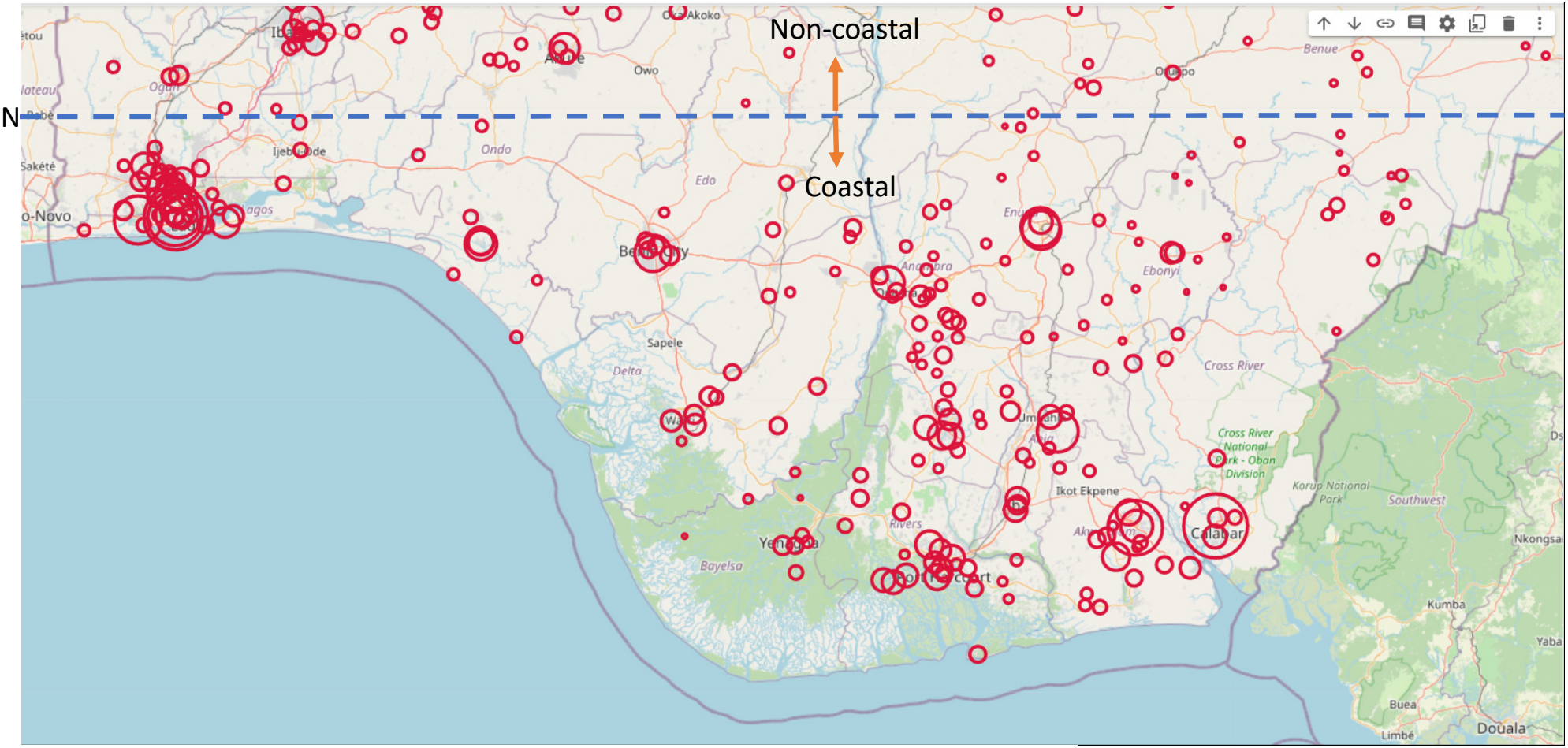
- Distance related features and day temperature not are not relevant in predicting median income!

Next Step: Lets identify new spatial features from locations of neighborhoods!

model-1

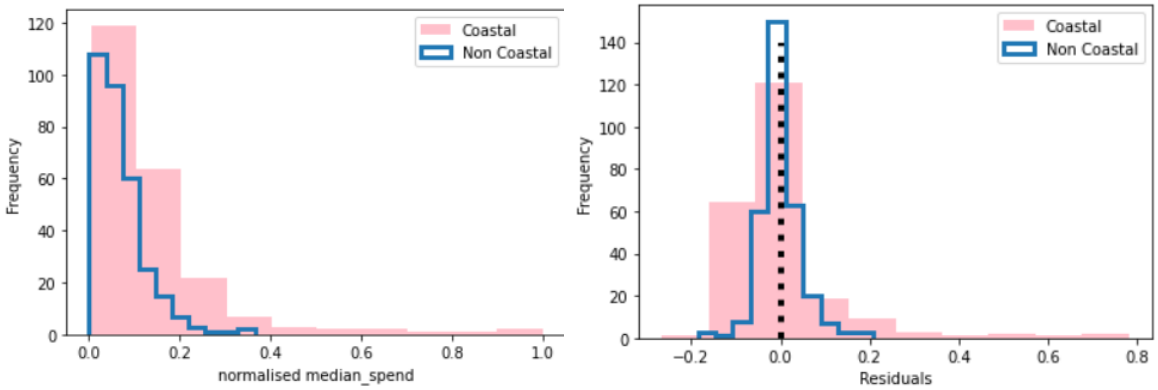
	Coeff.	Std. Error	P-Value
CONSTANT	-0.001874	0.070539	0.978818
accessibility_to_cities_2015	-0.115711	0.060745	0.057345
chirps_2015	0.226170	0.122887	0.066262
chirps_average_2002_2015	-0.172157	0.154479	0.265603
distance_to_ports_2012	0.023362	0.043499	0.591453
distance_to_powerplants_2016	-0.041558	0.034637	0.230763
distance_to_roads_2015	-0.021768	0.073081	0.765923
distance_to_transmission_lines_2016	-0.012770	0.032345	0.693139
landscan_population_2017	0.015428	0.067436	0.819126
modis_evi_2000_2016	0.151341	0.108327	0.162983
modis_lst_day_average_2015	-0.028958	0.051584	0.574784
modis_lst_night_average_2015	0.169921	0.051105	0.000946
modis_ndvi_2000_2016	-0.210679	0.119641	0.078834
sedac_gpw_2015	0.098950	0.054886	0.071988
srtm_2000	0.043541	0.046230	0.346713
viirs_nightlights_2015	0.254315	0.065395	0.000114

Identifying coastal neighborhoods




```
Coastal neighborhood samples: 223; non-coastal neighborhood samples: 318
Coastal neighborhood:non-coastal neighborhood = 0.41:0.59
------------------------------------------------------------------------------
The dataset is imbalanced in coastal feature
```

Influence of Coastal feature



Statistic= 0.31, p-value=0.75620272, f-value = 0.096

- Coastal neighborhoods' median expenditure is comparatively higher than the non-coastal ones.
- The error in coastal neighborhood is also higher than non-coastal ones with higher variances.
- Coastal feature does not add much significance in prediction. Adj. R2 reduces in model-2!

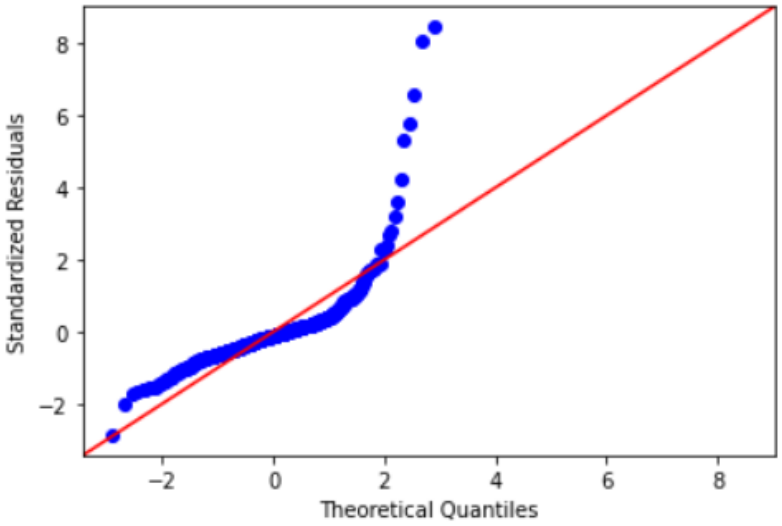
model-2

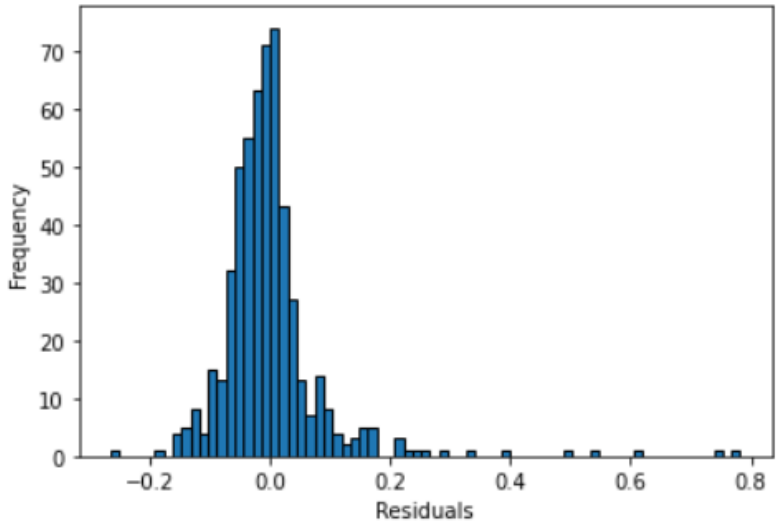
	Coeff.	Std. Error	P-Value
CONSTANT	-0.015269	0.073489	0.835486
accessibility_to_cities_2015	-0.113699	0.060856	0.062274
chirps_2015	0.242729	0.125533	0.053702
chirps_average_2002_2015	-0.207876	0.163927	0.205325
distance_to_ports_2012	0.025357	0.043630	0.561372
distance_to_powerplants_2016	-0.044624	0.034972	0.202523
distance_to_roads_2015	-0.020606	0.073143	0.778265
distance_to_transmission_lines_2016	-0.009361	0.032780	0.775319
landscan_population_2017	0.016629	0.067498	0.805494
modis_evi_2000_2016	0.145219	0.108790	0.182501
modis_lst_day_average_2015	-0.020755	0.053114	0.696133
modis_lst_night_average_2015	0.175477	0.051834	0.000764
modis_ndvi_2000_2016	-0.197325	0.121435	0.104775
sedac_gpw_2015	0.093977	0.055440	0.090645
srtm_2000	0.053542	0.048717	0.272252
viirs_nightlights_2015	0.253156	0.065455	0.000124
coastal	0.011275	0.017238	0.513343

	R2	Adj. R2
M1	0.304924	0.285065
M2	0.305491	0.284285

Anova Model (Double check)

	df	sum_sq	mean_sq	F	PR(>F)
accessibility_to_cities_2015	1.0	0.029828	0.029828	3.490660	0.062274
chirps_2015	1.0	0.031948	0.031948	3.738769	0.053702
chirps_average_2002_2015	1.0	0.013741	0.013741	1.608080	0.205325
distance_to_ports_2012	1.0	0.002886	0.002886	0.337765	0.561372
distance_to_powerplants_2016	1.0	0.013913	0.013913	1.628161	0.202523
distance_to_roads_2015	1.0	0.000678	0.000678	0.079369	0.778265
distance_to_transmission_lines_2016	1.0	0.000697	0.000697	0.081551	0.775319
landscan_population_2017	1.0	0.000519	0.000519	0.060697	0.805494
modis_evi_2000_2016	1.0	0.015226	0.015226	1.781858	0.182501
modis_lst_day_average_2015	1.0	0.001305	0.001305	0.152693	0.696133
modis_lst_night_average_2015	1.0	0.097934	0.097934	11.460773	0.000764
modis_ndvi_2000_2016	1.0	0.022563	0.022563	2.640454	0.104775
sedac_gpw_2015	1.0	0.024554	0.024554	2.873464	0.090645
srtm_2000	1.0	0.010322	0.010322	1.207907	0.272252
viirs_nightlights_2015	1.0	0.127825	0.127825	14.958801	0.000124
coastal	1.0	0.003656	0.003656	0.427830	0.513343
Residual	524.0	4.477638	0.008545	NaN	NaN





Coastal feature is not adding much significance in the model

Lets take a detour and look for other features!

Motivations: Try to unveil spatially patterned intrinsic features:
Spatial feature engineering

Exploring other Datasets

1. Admin buildings
2. Bank buildings

Find *poi* Categories

```
bank           301
admbuilding     46
Name: poi_type, dtype: int64
```

Check for uniqueness of bank buildings

```
data_admin_bank = data_admin[data_admin['poi_type']=='bank'].iloc[:, 1:]
data_admin_bank.reset_index(inplace = True, drop = True)
print('-------------------------------------------------------------------------------')
print('## Check if poi = "Bank" for bank data and Admin building are same?!? ##')
if data_banks.values.tolist()==data_admin_bank.values.tolist():
  print('Bank Data are same: Consider only the Admin bdg table')
else:
  print('Bank Data are not same: Consider both the tables')
print('-------------------------------------------------------------------------------')
```

```
-------------------------------------------------------------------------------
## Check if poi = "Bank" for bank data and Admin building are same?!? ##
Bank Data are same: Consider only the Admin bdg table
-------------------------------------------------------------------------------
```

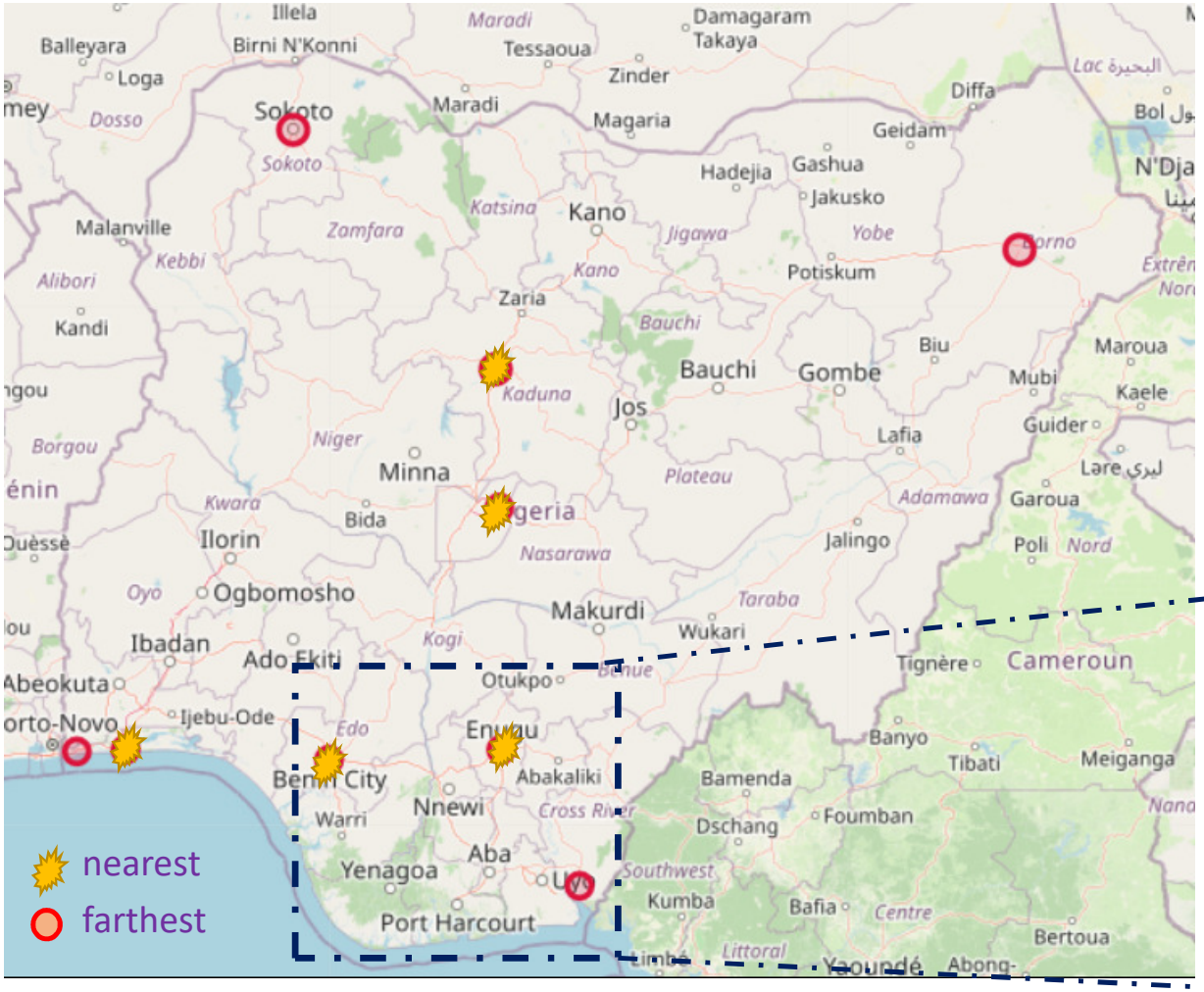
Let's consider admin building dataset only!

Location of administrative buildings

Motivations: Try to locate a bunch of secluded and urban neighborhoods for better spatial zoning

Approach:

1. Get distances between locations (considers earth curvature)
2. Get the Distance matrix
3. Extract the 5 most nearest and farthest neighborhoods for each location.
4. Plot the neighborhoods



Approach:

4. Plot the neighborhood locations (total = 9)

```
Most common neighborhood with admin buildings:
poi_type                                  bank
lon                                       7.49
lat                                       6.44
name        First City Monument Bank (FCMB)
```

Most significant neighborhood

Calculating distances from each Nigerian neighborhood to FCMB building

```
eu_dist_comm = []
for k, n in data[['lat','long' ]].itertuples(index=False):
  eu_dist_com = euclid_dist(common_loc[1], common_loc[2], n, k)
  eu_dist_comm.append(eu_dist_com)
len(eu_dist_comm)
data['dist_FCMB'] = eu_dist_comm
data.head()
```

dist_FCMB
112.085906
108.725589
198.325633
73.817541
128.266087

- Spatial feature does add some significance in prediction. Adj. R2 increase in model-3!

	R2	Adj. R2
M1	0.304924	0.285065
M2	0.305491	0.284285
M3	0.311318	0.290289

model-3

	Coeff.	Std. Error	P-Value
CONSTANT	-0.028769	0.071331	0.686875
accessibility_to_cities_2015	-0.126335	0.060714	0.037937
chirps_2015	0.272360	0.124215	0.028771
chirps_average_2002_2015	-0.194931	0.154260	0.206917
distance_to_ports_2012	0.007629	0.043923	0.862173
distance_to_powerplants_2016	-0.051839	0.034824	0.137193
distance_to_roads_2015	-0.036447	0.073117	0.618361
distance_to_transmission_lines_2016	-0.011055	0.032236	0.731777
landscan_population_2017	0.017053	0.067193	0.799752
modis_evi_2000_2016	0.106051	0.109867	0.334854
modis_lst_day_average_2015	-0.055816	0.052818	0.291106
modis_lst_night_average_2015	0.176358	0.051002	0.000589
modis_ndvi_2000_2016	-0.159817	0.121413	0.188648
sedac_gpw_2015	0.079198	0.055414	0.153540
srtm_2000	0.045805	0.046072	0.320578
viirs_nightlights_2015	0.251525	0.065168	0.000128
dist_FCMB	0.080486	0.036492	0.027845

Spatial Feature Engineering Works!

Lets discover more on Spatial Heterogeneity from Neighborhood Clustering

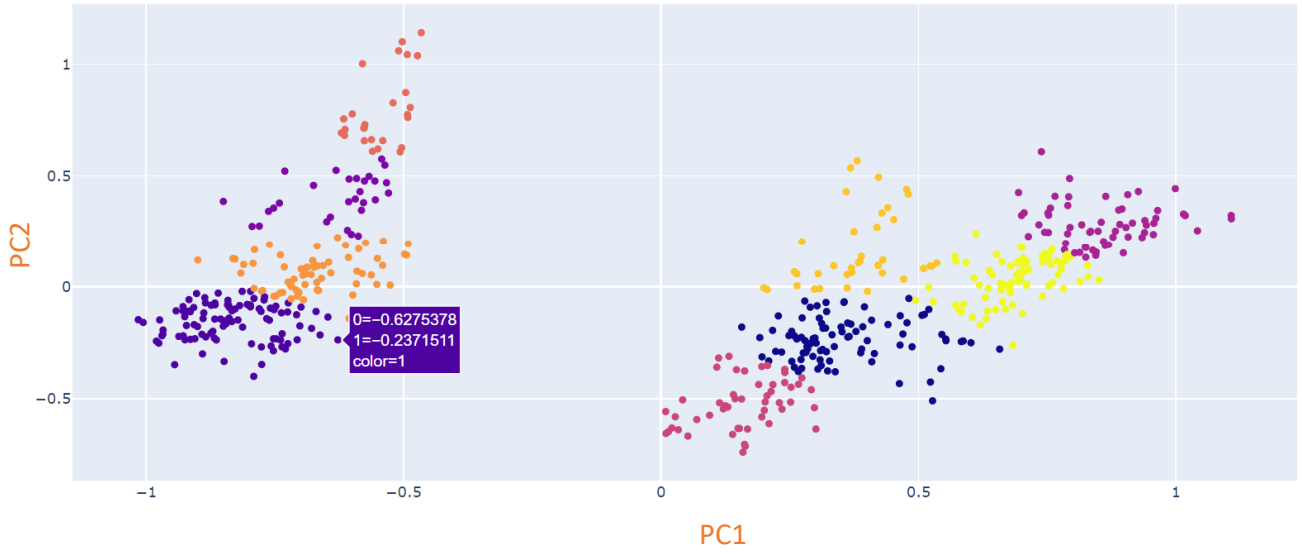
Some neighborhoods might be have more expenditure than other neighborhoods due to un-modeled/unidentified latent features/locations!

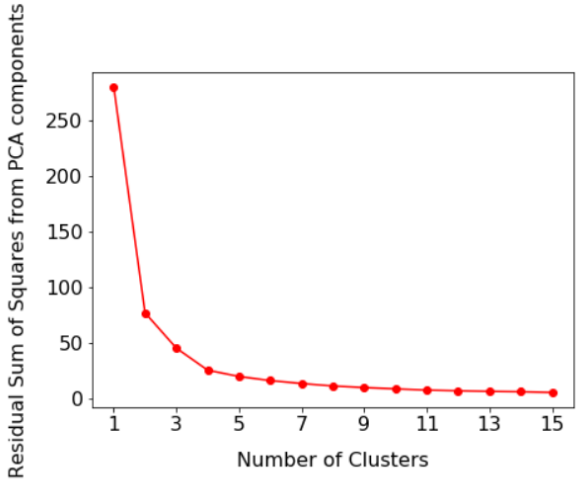
We need to discover that?!?

Clustering:: on 1st two PCA components

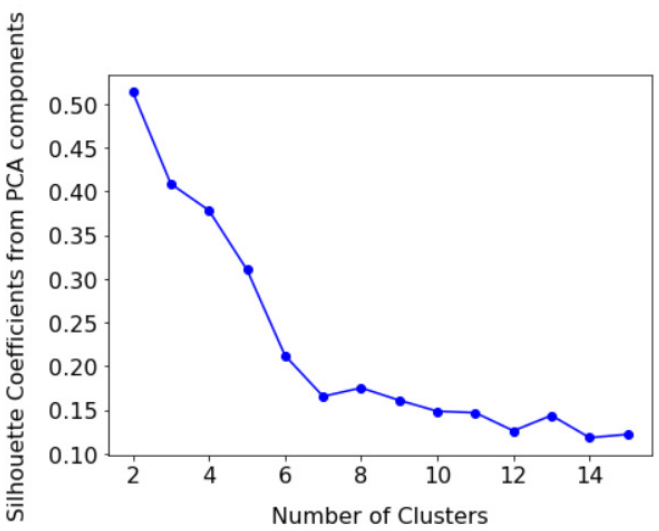
```
Variance Explained: [0.62163134 0.15161037]
Total variance explained: 77.32%
```

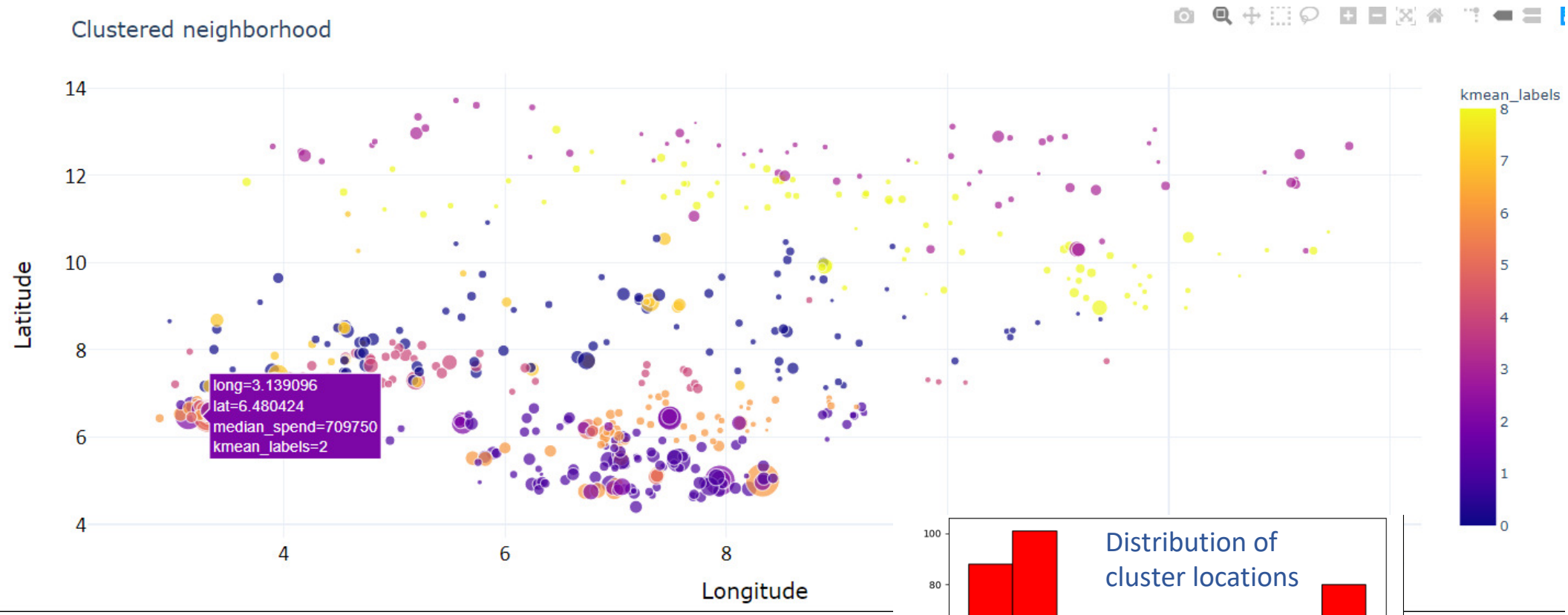
Considered total #of Clusters = 9

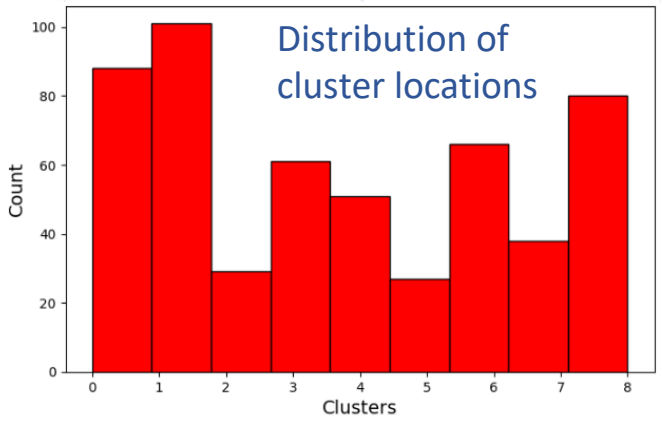


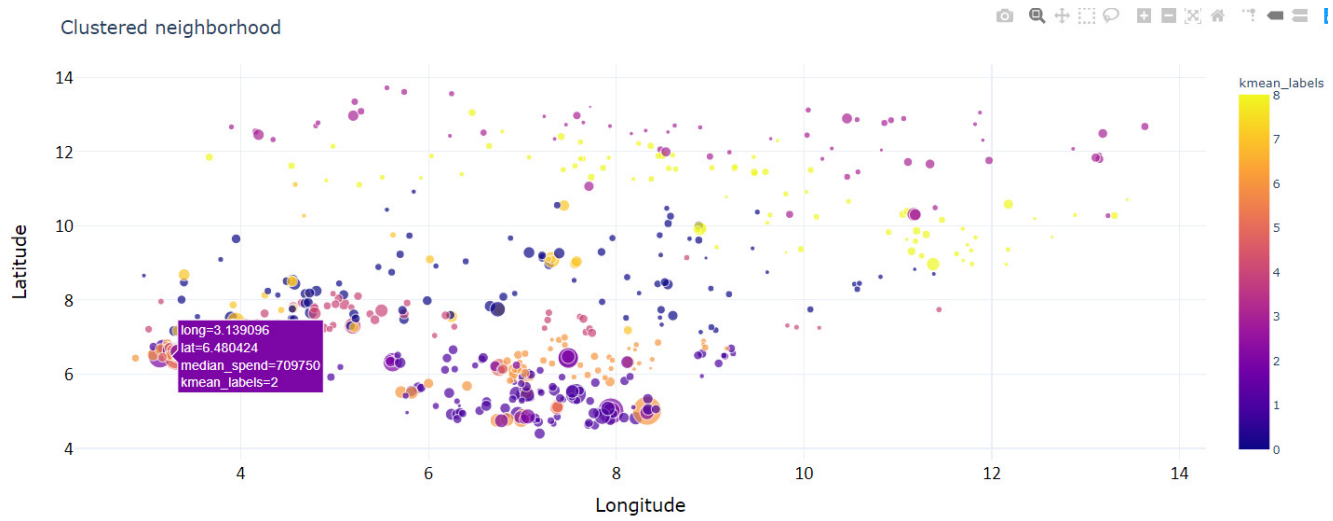


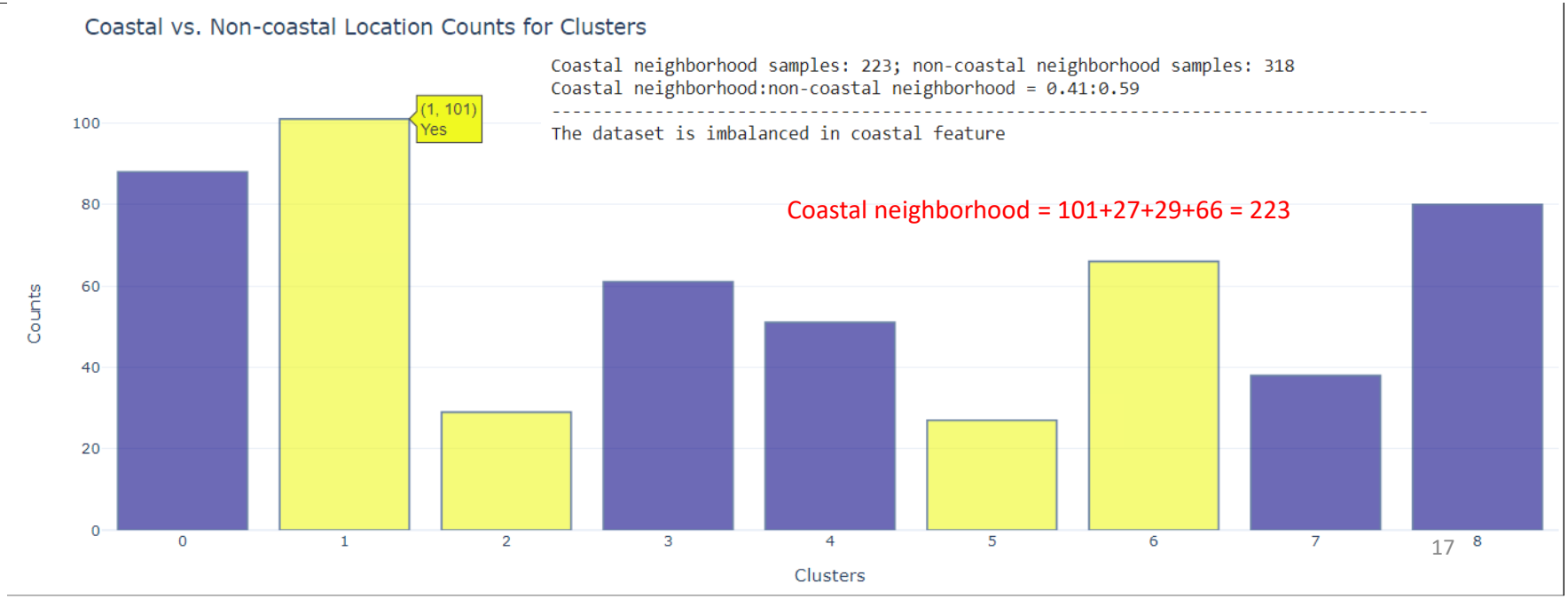

```
Optimal no. of clusters from kmeans: 4
```



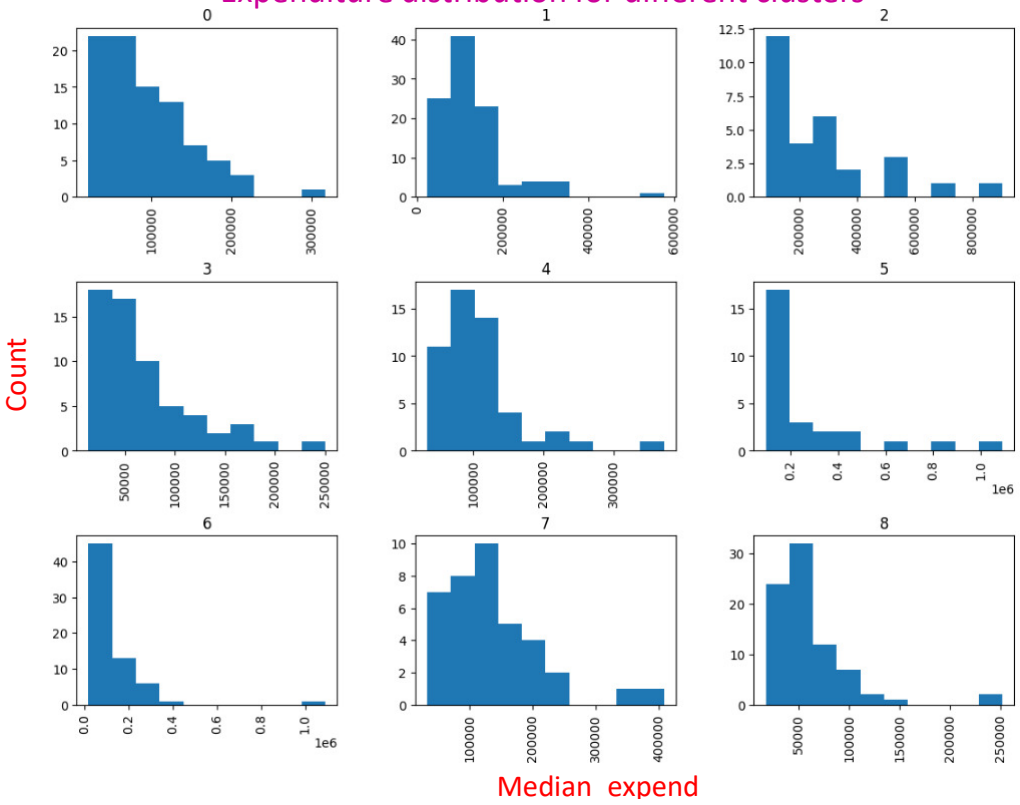

Clustered neighborhood
Latitude
Longitude
kmean_labels
long=3.139096
lat=6.480424
median_spend=709750
kmean_labels=2

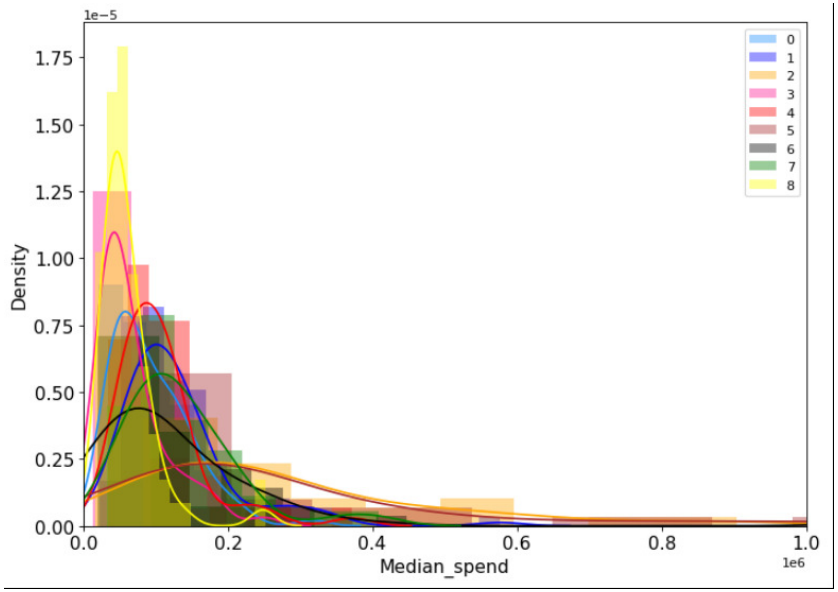

Distribution of cluster locations
Count
Clusters


Clustered neighborhood
Latitude
Longitude
kmean_labels
long=3.139096
lat=6.480424
median_spend=709750
kmean_labels=2


Coastal vs. Non-coastal Location Counts for Clusters
Coastal neighborhood samples: 223; non-coastal neighborhood samples: 318
Coastal neighborhood:non-coastal neighborhood = 0.41:0.59
The dataset is imbalanced in coastal feature
Coastal neighborhood = 101+27+29+66 = 223
(1, 101)
Yes
Counts
Clusters

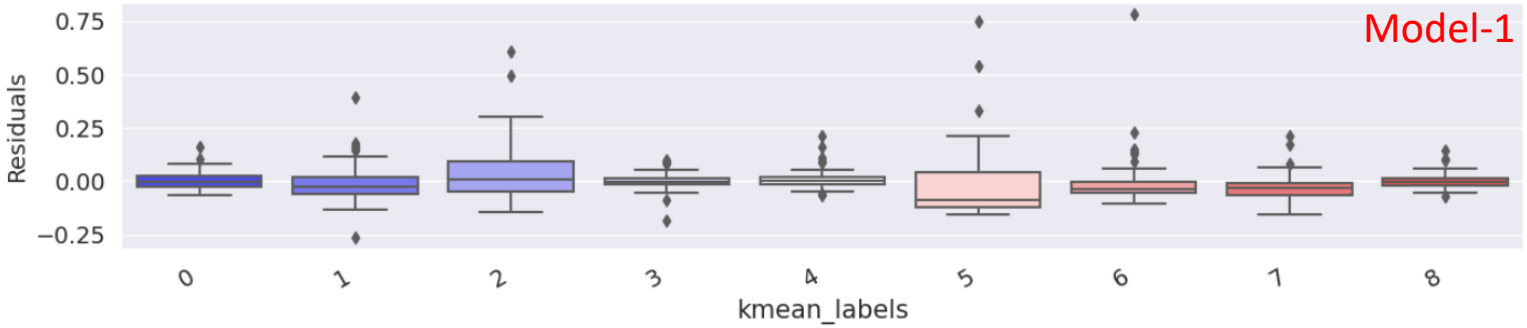
Expenditure distribution for different clusters





- Coastal clusters have higher expenditure spend than the rests!

- Coastal clusters are subjected to higher expenditure residuals



Model-1

Median variations of different attributes for different clusters

kmean_labels	0	1	2	3	4	5	6	7	8
accessibility_to_cities_2015	1.710	1.707	0.305	1.600	1.700	0.300	1.500	0.481	2.180
chirps_2015	3.321	5.609	4.369	1.687	3.613	3.424	4.578	3.169	2.475
chirps_average_2002_2015	1280.776	2408.844	1914.559	631.653	1475.839	1525.226	1979.234	1222.723	930.274
distance_to_ports_2012	14.200	5.833	5.100	25.596	11.296	2.621	7.119	8.597	22.300
distance_to_powerplants_2016	1.345	0.335	0.293	3.666	1.084	0.235	0.699	0.859	2.882
distance_to_roads_2015	514.505	377.380	112.058	597.120	428.708	27.766	316.302	62.087	547.354
distance_to_transmission_lines_2016	0.081	0.042	0.005	0.025	0.075	0.000	0.044	0.026	0.033
landscan_population_2017	127.383	547.445	6178.262	307.647	351.698	17116.523	1314.328	7001.192	197.972
modis_evi_2000_2016	3325.326	4078.636	2568.576	2004.568	4162.808	1700.982	3519.462	2410.545	2462.091
modis_lst_day_average_2015	33.505	28.339	31.730	40.523	29.769	33.722	29.930	35.863	37.959
modis_lst_night_average_2015	19.640	20.397	21.739	18.272	19.577	22.460	20.469	20.795	17.834
modis_ndvi_2000_2016	4713.766	5497.063	3522.778	2687.028	5844.230	2295.782	4766.041	3419.996	3440.352
sedac_gpw_2015	147.398	544.709	2030.041	213.422	289.326	11553.672	639.350	1159.490	187.761
srtm_2000	309.458	62.269	38.071	367.650	344.383	19.447	69.145	300.385	474.208
viirs_nightlights_2015	0.000	0.005	7.381	0.000	0.000	11.833	0.009	3.291	0.000

Spatial Heterogeneity : Fixed Effects

- Fit the statistical model w/o intercept to verify appropriate cluster allocations.
- Neighborhood locations are modeled using binary variables.
- Fixed effect is considerably small which indicates locations from an assigned cluster belongs to the same cluster.

	Coef.	Std. Error	P-Value
kmean_labels[0]	-1.997423e-16	5.792110e-16	0.730347
kmean_labels[1]	-7.816948e-17	6.132109e-16	0.898614
kmean_labels[2]	-3.929258e-16	5.533874e-16	0.478004
kmean_labels[3]	-3.128201e-16	6.432901e-16	0.626975
kmean_labels[4]	-1.080326e-16	5.981473e-16	0.856743
kmean_labels[5]	-5.668423e-16	5.342010e-16	0.289142
kmean_labels[6]	9.230095e-17	5.692053e-16	0.871246
kmean_labels[7]	-1.601163e-16	5.636414e-16	0.776468
kmean_labels[8]	-2.382398e-16	6.113472e-16	0.696923

Fixed effects = (data+FCMB_dist)

$$P_i = \alpha_r + \sum_k \mathbf{X}_{ik}\beta_k + \epsilon_i$$

- The constant term, α, varies by kmeans labels r, α_r
- Intercept term is cluster specific

- P-values are not significant thus spatial importance/fixed effect from sm.ols() model is not valuable for spatial Heterogeneity.

Spatial Heterogeneity : Spatial Regimes (kmeans-clustering)

```
# spatial regimes implementation
m5 = spreg.OLS_Regimes(
    y = data_FCMB[['median_spend']].values,
    x =data_FCMB[data_FCMB.columns[3:].tolist()].values,
    regimes= data_FCMB['kmean_labels'].tolist(),
    constant_regi='many',
    regime_err_sep=True,
    name_y='median_spend',
    name_x=data_FCMB.columns[3:].tolist()
)
```

- Spatial Regimes on k-means labels

$$P_i = \alpha_r + \sum_k \mathbf{X}_{ki}\beta_{k-r} + \epsilon_i$$

α_r, β_{k-r} are varying terms

- Intercept and coeff. are cluster specific

	Coeff
0_kmean_label	-0.009494
1_kmean_label	-0.099047
2_kmean_label	0.109589
3_kmean_label	-0.294385
4_kmean_label	-0.031606
5_kmean_label	5.521021
6_kmean_label	0.819976
7_kmean_label	0.491507
8_kmean_label	-0.218357

OLS_regimes (data+FCMB_dist)

- Coeffs of few of the clusters are significant! Which means clustering plays role in spatial heterogeneity.
- Significant improvement has been made from initial spatial regression model (chow-test, p-values).

Chow-test (OLS_regimes)

	Statistic	P-value
CONSTANT	4.507526	8.086796e-01
accessibility_to_cities_2015	18.022333	2.105968e-02
chirps_2015	45.386485	3.108507e-07
chirps_average_2002_2015	27.988459	4.764480e-04
distance_to_ports_2012	4.022651	8.550741e-01
distance_to_powerplants_2016	6.313503	6.121597e-01
distance_to_roads_2015	10.875667	2.088441e-01
distance_to_transmission_lines_2016	7.522859	4.814076e-01
landscan_population_2017	3.007371	9.338942e-01
modis_evi_2000_2016	3.481761	9.006006e-01
modis_lst_day_average_2015	8.256357	4.088376e-01
modis_lst_night_average_2015	8.694684	3.687030e-01
modis_ndvi_2000_2016	3.495931	8.995052e-01
sedac_gpw_2015	10.261280	2.471643e-01
srtm_2000	3.787459	8.757724e-01
viirs_nightlights_2015	13.080297	1.091165e-01
dist_FCMB	3.518038	8.977852e-01

Spatial Heterogeneity : Spatial Regimes (coastal spatial feature)

- Spatial Regimes on coastal tags

$$P_i = \alpha_r + \sum_k \mathbf{X}_{ki}\beta_{k-r} + \epsilon_i$$

- Coeffs of coastal and noncoastal neighborhoods vary significantly, making difficult to establish any relationship!

- P-values are quite higher than 0.05; which symbolizes coastal feature does not contribute in spatial heterogeneity to predict median expenditure!

Coefficients (OLS_regime)

	Coastal Coeff.	Non-coastal Coeff.
CONSTANT	-0.515001	0.066253
accessibility_to_cities_2015	-0.171515	-0.108590
chirps_2015	0.516448	-0.024850
chirps_average_2002_2015	-0.475892	0.054128
distance_to_ports_2012	0.156840	-0.030065
distance_to_powerplants_2016	0.152198	-0.012253
distance_to_roads_2015	0.007315	-0.075472
distance_to_transmission_lines_2016	-0.305781	-0.011537
landscan_population_2017	-0.058524	0.116814
modis_evi_2000_2016	0.278409	NaN
modis_lst_day_average_2015	0.033065	-0.055796
modis_lst_night_average_2015	0.682188	0.088139
modis_ndvi_2000_2016	-0.220091	NaN
sedac_gpw_2015	0.135629	-0.132340
srtm_2000	-0.015752	0.041199
viirs_nightlights_2015	0.186376	0.226605
dist_FCMB	0.027420	0.017508

Chow-test (OLS_regimes)

	Statistic	P-value
CONSTANT	6.556383	0.010451
accessibility_to_cities_2015	0.125069	0.723601
chirps_2015	4.262122	0.038971
chirps_average_2002_2015	1.975910	0.159822
distance_to_ports_2012	1.702735	0.191931
distance_to_powerplants_2016	0.413753	0.520071
distance_to_roads_2015	0.307405	0.579277
distance_to_transmission_lines_2016	1.671956	0.195997
landscan_population_2017	1.704320	0.191724
modis_evi_2000_2016	0.137076	0.711205
modis_lst_day_average_2015	0.373670	0.541010
modis_lst_night_average_2015	8.045060	0.004563
modis_ndvi_2000_2016	0.006353	0.936473
sedac_gpw_2015	5.818237	0.015861
srtm_2000	0.104209	0.746836
viirs_nightlights_2015	0.099311	0.752659
dist_FCMB	0.007277	0.932017

Now Lets do some Maths !?!

WHY??

It helps to interpret the neighborhood connectivity better

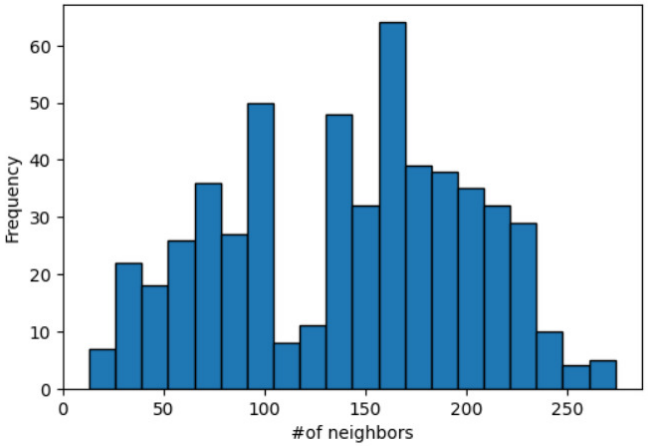
Weight Matrix formation from distances matrix (block weights)

Approach:

1. Normalized the distance matrix by indices (/rows)
2. Filter the values > 0.2 =0 else 1 [0.2 coeff. = hyperparameter]
3. Replace the diagonals back to zero!

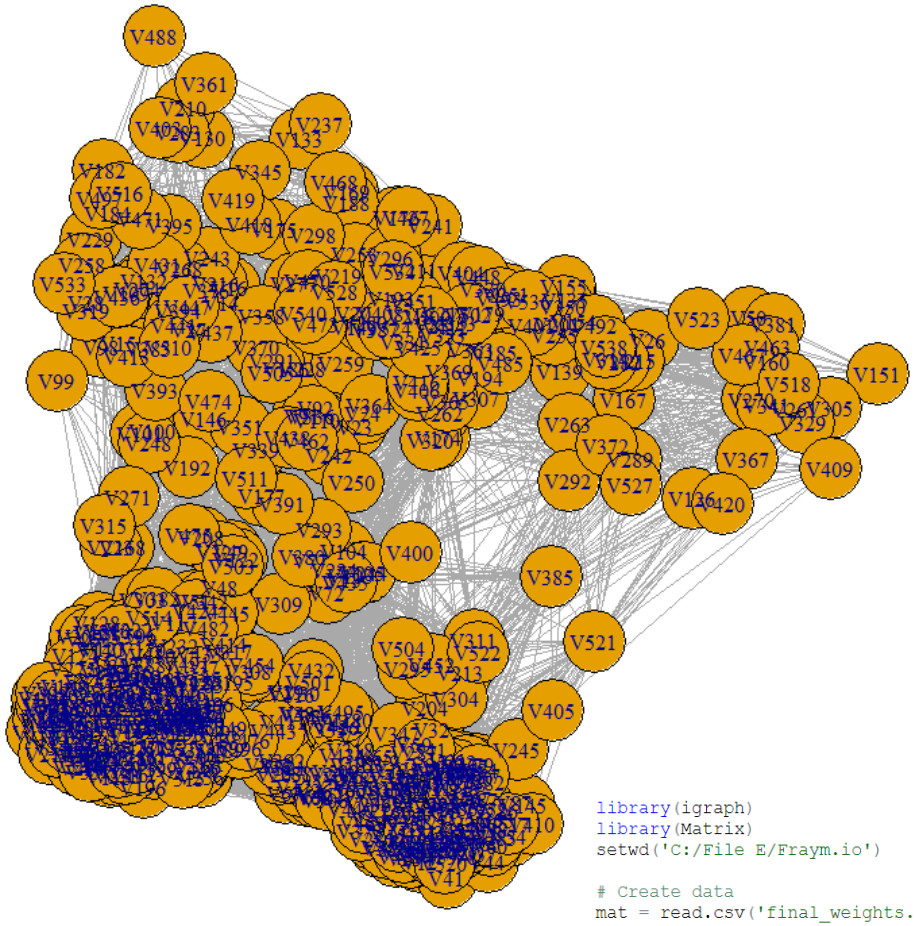
```
eu_wt_mat /= eu_wt_mat.max(axis=(0, 1))
# print(eu_wt_mat)
filtered = []
for i in eu_wt_mat:
  for j in i:
    if j>0.2:
      j =0
    else:
      j=1
    filtered.append(j)
final_wt = np.reshape(filtered, (len(data), len(data)))
final_wt[np.diag_indices_from(final_wt)] = 0
```

Distribution of Neighbors



1st row of Weight Matrix

```
matrix([[0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1,
         0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
         1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1,
         1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
         0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
         1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0,
         0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0,
         1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0,
         0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1,
         1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0,
         0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0,
         1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0,
         0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1,
         0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1,
         0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0,
         1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1,
         0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0,
         1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1,
         0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0,
         0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0,
         1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0,
         0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0,
         1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0,
         0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
         1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1]])
```



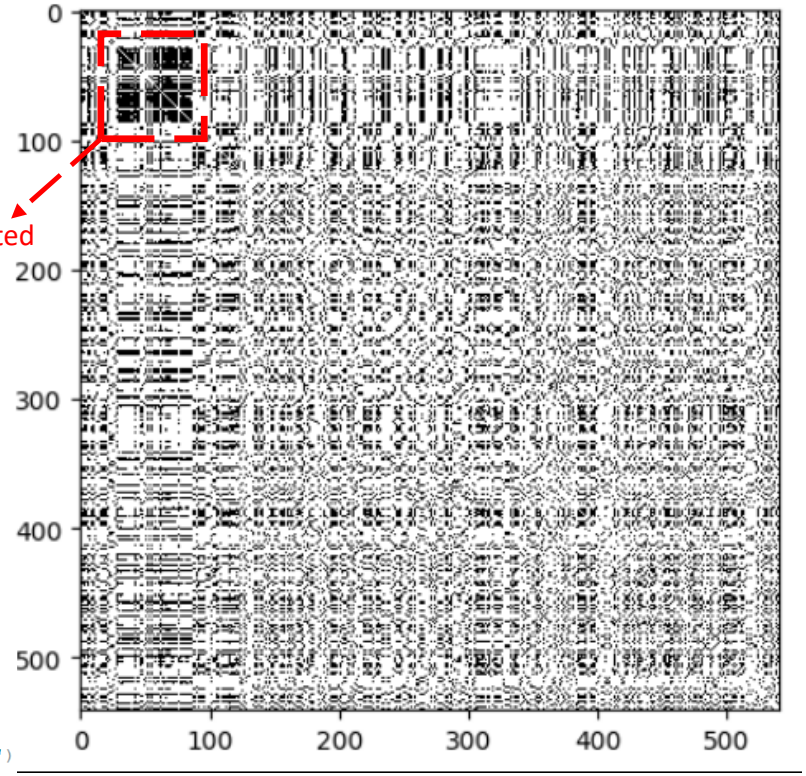
Weight Matrix Plot (igraph)

```
library(igraph)
library(Matrix)
setwd('C:/File E/Fraym.io')

# Create data
mat = read.csv('final_weights.csv',
               header = FALSE, sep = ",", skip = 0)
mat = as.matrix(mat)
network <- graph_from_adjacency_matrix(mat ,
                                       mode='undirected')
plot(network, layout=layout.fruchterman.reingold,
     main="fruchterman.reingold")
```

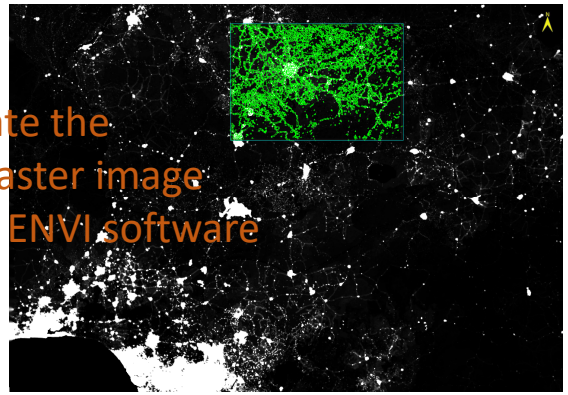
Weight Matrix

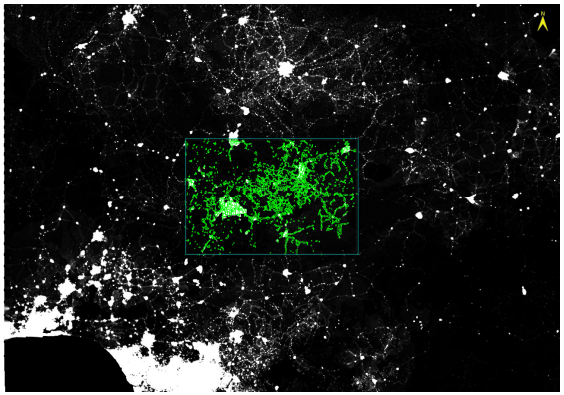
Closely connected locations



Neighborhood Detection from Man-made Imperviousness (Raster data:: 2010) from ENVI

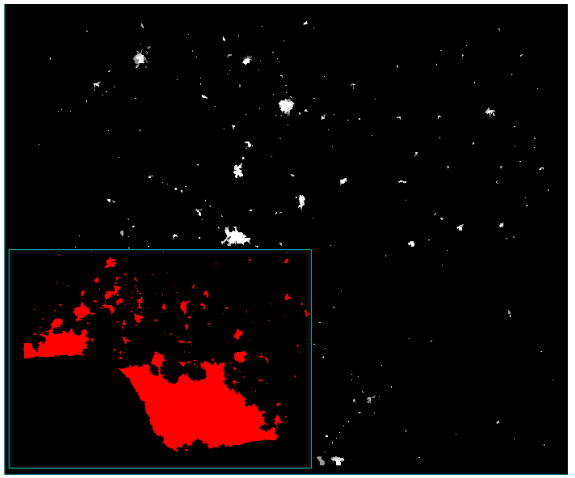
Approach:
1. To generate the shape.file, raster image imported in ENVI software

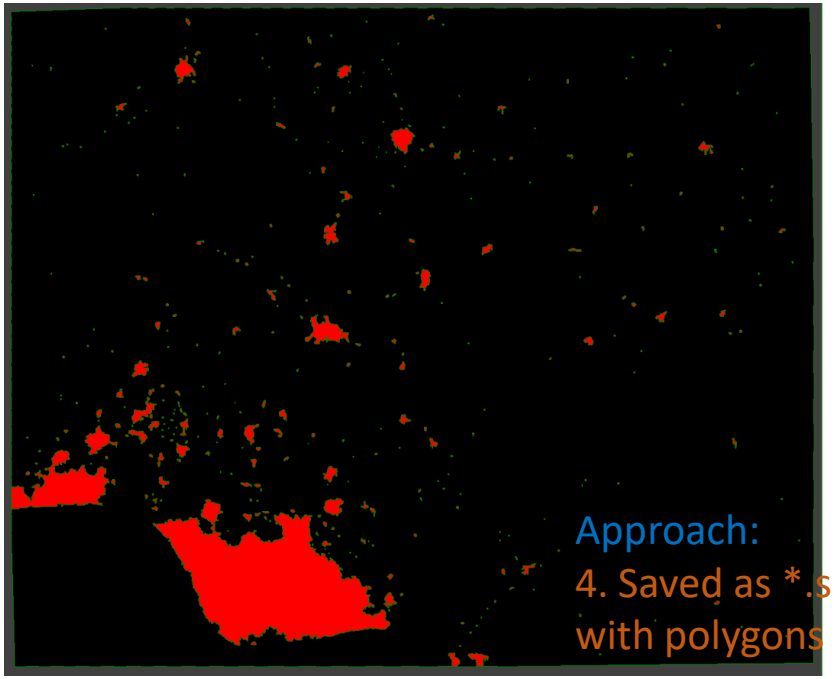




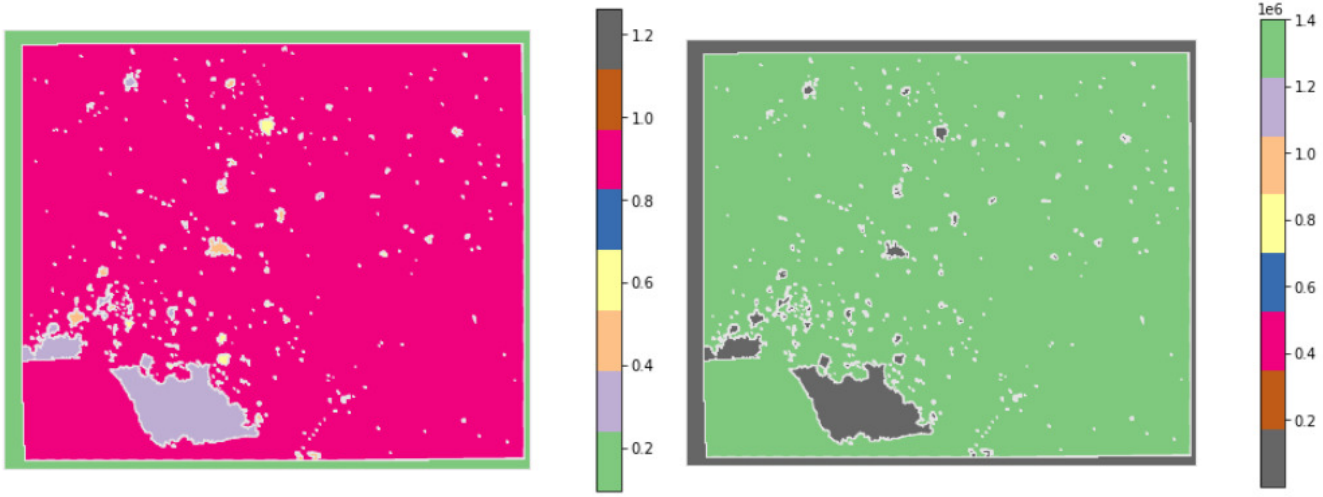
Approach:
2. Rule-based feature extraction (Edge Segment scale=80, merge scale = 20)

Approach:
3. filtering Spectral mean frequency [0.06-90]





Approach:
4. Saved as *.shp with polygons

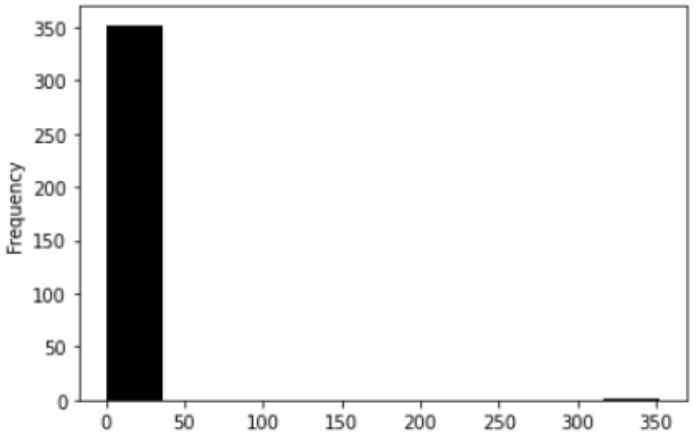


FX_ROUND

Area

```
shapefile Reader
    353 shapes (type 'POLYGON')
    353 records (26 fields)

        {'AREA': 6.0,
         'AVG_B1': 3.369303,
         'CLASS_ID': '1',
         'CLASS_NAME': 'New Class 1',
         'FX_AREA': 5995400.5,
         'FX_COMPACT': 0.27616,
         'FX_CONVEX': 1.0,
         'FX_ELONG': 1.512691,
         'FX_FORMFAC': 0.7527,
         'FX_HOLESOL': 1.0,
         'FX_LENGTH': 10004.676758,
         'FX_MAIN_DI': 180.0,
         'FX_MAJAXLN': 3011.509277,
         'FX_MINAXLN': 1990.829224,
         'FX_NUMHOLE': 0.0,
         'FX_RECT_FI': 1.0,
         'FX_ROUND': 0.841705,
         'FX_SOLID': 1.0,
         'MAX_B1': 4.704835,
         'MIN_B1': 2.152146,
         'STD_B1': 0.920317,
         'TXAVG_B1': 2.735511,
         'TXENT_B1': -0.589401,
         'TXRAN_B1': 3.545724,
         'TXVAR_B1': 1.526246}
```



Cardinality (#of neighbor)

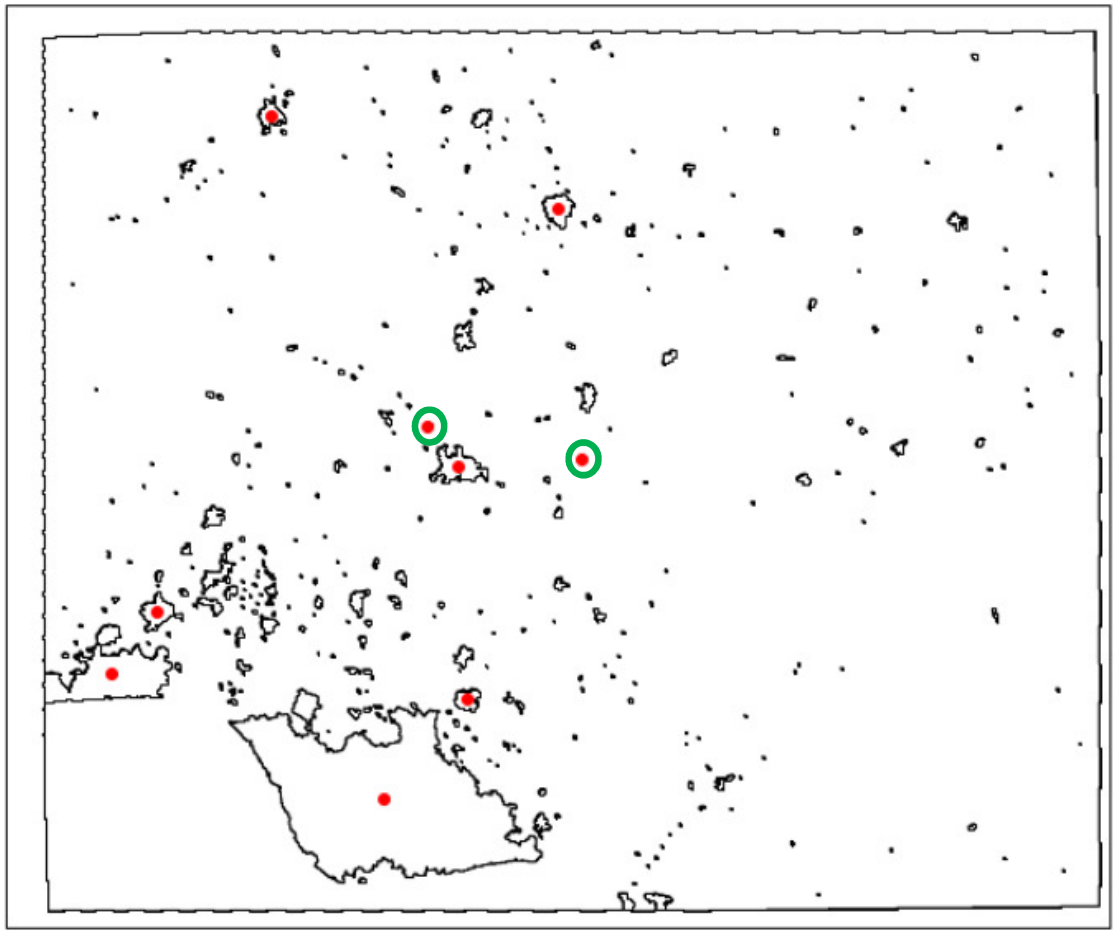
- Coarse segregation could detect highly populated neighborhood from foundation imperviousness
- Imbalanced Cardinality distribution
- Needs to increase the #of polygons

Polygons Centroid plot

Sorting the gdf by (descending) area (km^2)

	AREA	geometry	FX_ROUND	FX_CONVEX
351	1403523.0	POLYGON ((314884.844 443399.469, 294557.905 44...	0.969069	3.637995
352	148424.0	POLYGON ((1657479.166 421307.539, 239675.170 4...	0.096296	1.990236
345	48298.0	POLYGON ((674671.665 479561.481, 670606.277 47...	0.331432	2.216399
289	7981.0	POLYGON ((289476.170 710001.567, 288459.823 71...	0.356922	2.169446
158	1723.0	POLYGON ((808829.462 992038.839, 800698.686 99...	0.406016	1.961885

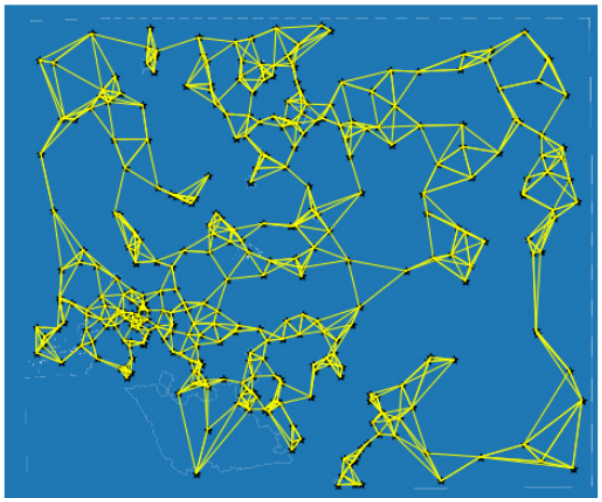
- As the polygons have been sorted by area, the non-impervious polygons get detected too!!



Plotting the centroids of first 9 polygons

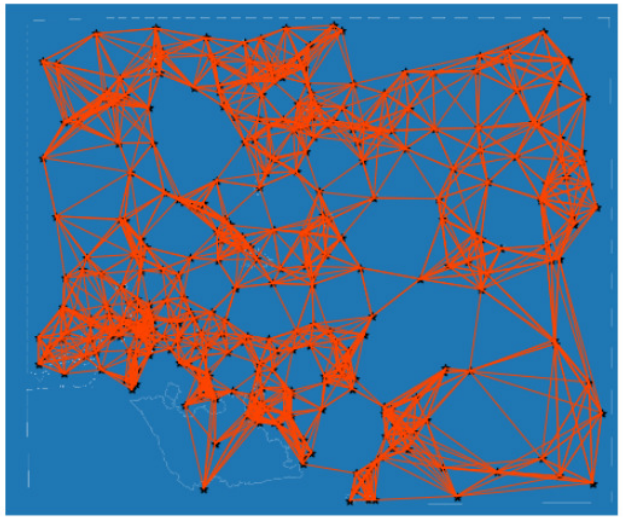
KNN Analysis

K-NN 4

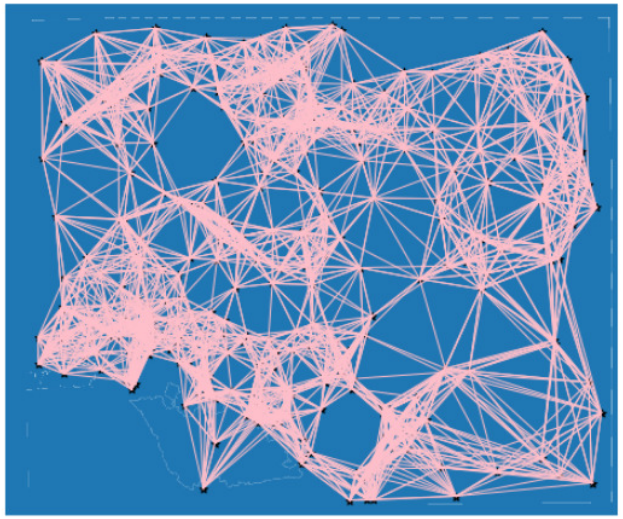


- KNN retrieves the dependence through the effect associated with spatial neighbors.
- KNN= 4,9,15 neighbors have been carried out to detect highly connected polygons.
- Network gets dense for k=15, capturing impervious polygons.
- To observe more dense connectivity, #of polygons should be increased.

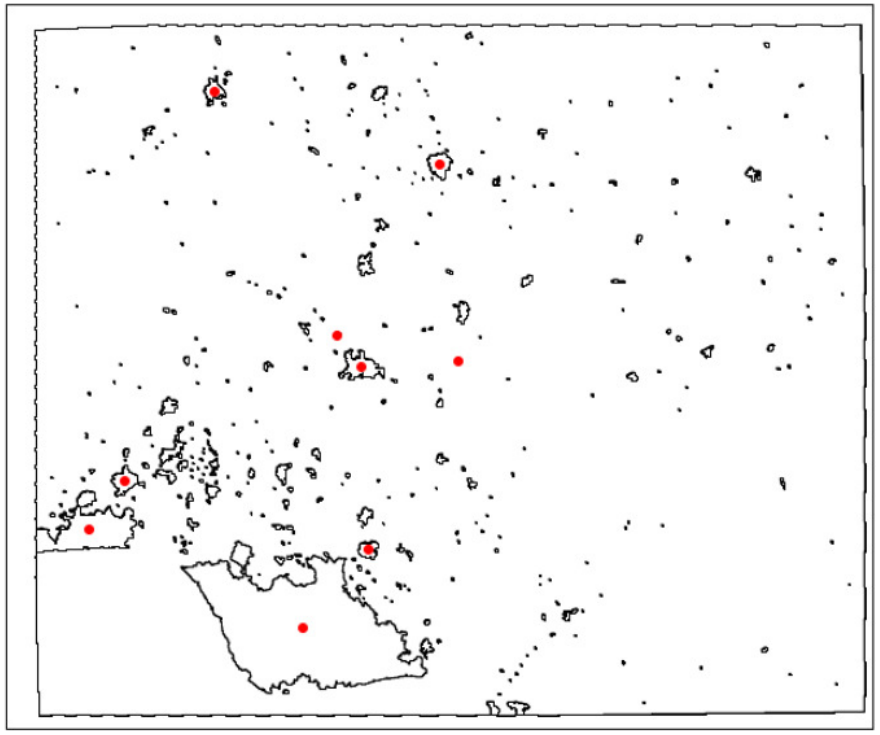
K-NN 9



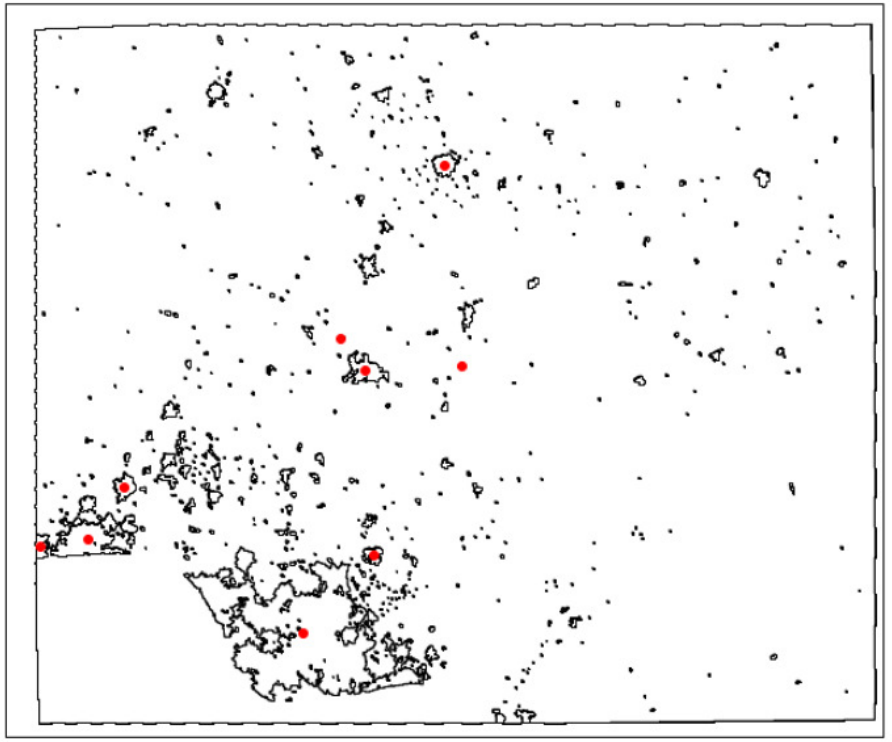
K-NN 15



Reiterations for #of records



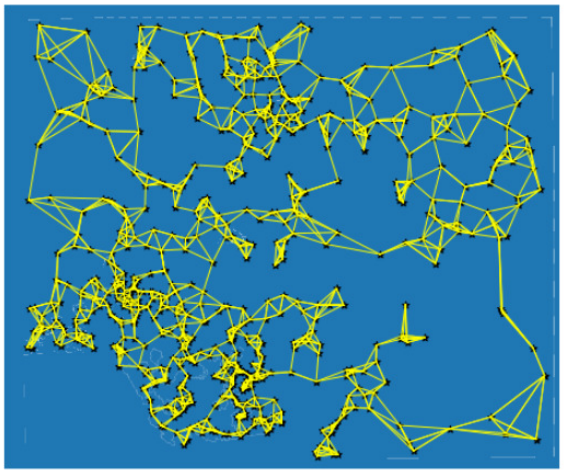
```
shapefile Reader
    353 shapes (type 'POLYGON')
    353 records (26 fields)
```



```
shapefile Reader
    676 shapes (type 'POLYGON')
    676 records (26 fields)
```

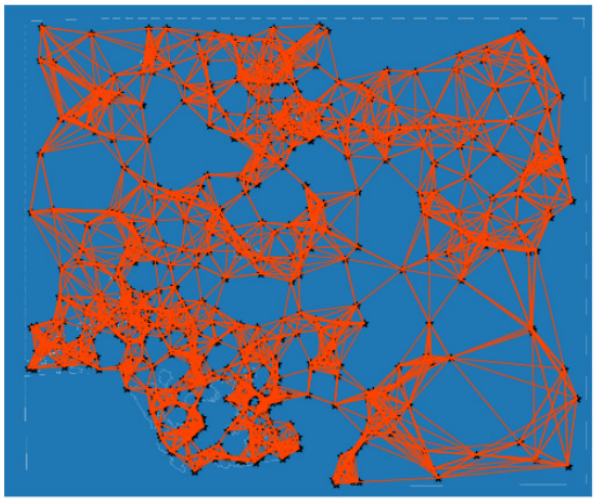
KNN Analysis (reiteration)

K-NN 4

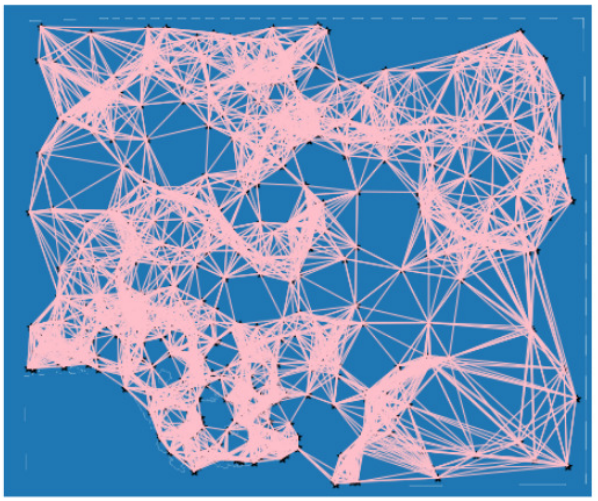


- Connectivity gets better forming more dense network for highly impervious regions.
- Highly impervious regions are compatible with highly connected neighborhood from weight matrix plot!
- Soil imperviousness and neighbor population are correlated!
- Densely connected neighborhood/highly populated region has higher median expenditure.

K-NN 9



K-NN 15



Part-2 :: Case Study on Nigerian Neighborhood to predict Median expenditure (spatial features included)

Algorithms		Hyper parameters	Part-1: Accuracy Matrices (%)		Part-2: Accuracy Matrices (%)	
			Train Set	Test Set	Train Set	Test Set
Polynomial Regression		--	48.9	42.16	--	--
Random Forest		500 trees	83.12	54.01	82.71	53.78
XGBoost		learning_rate= 0.01, max_depth= 2, n_estimators= 150	75.32	56.92	73.51	57.01
Decision Tree		max_depth = 7, n_estimators = 1	75.68	46.41	73.25	44.18
AdaBoost		max_depth = 7, n_estimators = 500	53.36	79.48	54.22	80.9
KNN		k=3/6	60.34	47.50	44.32	34.48
SVM	Linear	--	34.95	34.66	--	--
	Radial	--	34.91	34.62	--	--
	Poly.	Degree = 14	63.48	50.49	--	--

Scaling the expenditure to logarithmic form to minimize the skewness

spreg.OLS(Log(P) ~ .)

	Coeff.	Std. Error	P-Value
CONSTANT	10.189339	0.405372	5.185284e-92
accessibility_to_cities_2015	-1.909531	0.338926	2.879180e-08
chirps_2015	0.009823	0.694192	9.887151e-01
chirps_average_2002_2015	0.801477	0.860956	3.523272e-01
distance_to_ports_2012	0.196581	0.245617	4.238684e-01
distance_to_powerplants_2016	-0.591489	0.194830	2.517510e-03
distance_to_roads_2015	-0.591161	0.408260	1.482156e-01
distance_to_transmission_lines_2016	-0.103750	0.180432	5.655349e-01
landscan_population_2017	0.818034	0.375177	2.967324e-02
modis_evi_2000_2016	0.936160	0.614259	1.281016e-01
modis_lst_day_average_2015	-0.442184	0.294715	1.341199e-01
modis_lst_night_average_2015	1.610767	0.284668	2.519499e-08
modis_ndvi_2000_2016	-0.917682	0.677401	1.760948e-01
sedac_gpw_2015	-0.302763	0.309729	3.287692e-01
srtm_2000	0.732368	0.257240	4.586359e-03
viirs_nightlights_2015	1.515604	0.364337	3.721438e-05
dist_FCMB	0.612141	0.203710	2.783571e-03
kmean_labels	-0.150703	0.069719	3.110415e-02

Algorithms	Hyper parameters	Part-2: Accuracy Matrices (%) (Revised)	
		Train Set	Test Set
Polynomial Regression	--	--	--
Random Forest	500 trees	98.88	96.94
XGBoost	learning_rate= 0.01, max_depth= 2, n_estimators= 150	98.17	96.93
Decision Tree	max_depth = 7, n_estimators = 1	98.53	95.71
AdaBoost	max_depth = 7, n_estimators = 500	96.75	99.67
KNN	k=6	59.6	52.47

	R2	Adj. R2
M1	0.304924	0.285065
M2	0.305491	0.284285
M3	0.311318	0.290289
M7	0.495973	0.479590

Summary

- The predictor variable is imbalanced (/skewed) which makes the statistical models instable.
- Coastal feature does not add any spatial importance in predicting median expenditure.
- Distance from most connected neighborhood adds spatial importance.
- K-means clustering (k=9) is substantiating spatial heterogeneity.
- Block weight matrix corroborate spatial locations of densely connected neighborhood.
- Highly impervious regions are compatible with highly connected neighborhood from weight matrix plot!
- Densely connected neighborhood has higher comparatively higher median expenditure.
- Conventional ML algorithms performance did not improve much after considering spatial features until logarithmic expenditure is used for prediction.
- More complex structure like artificial neural network could be considered to explore more but model interpretations might reduce!

Thank You!