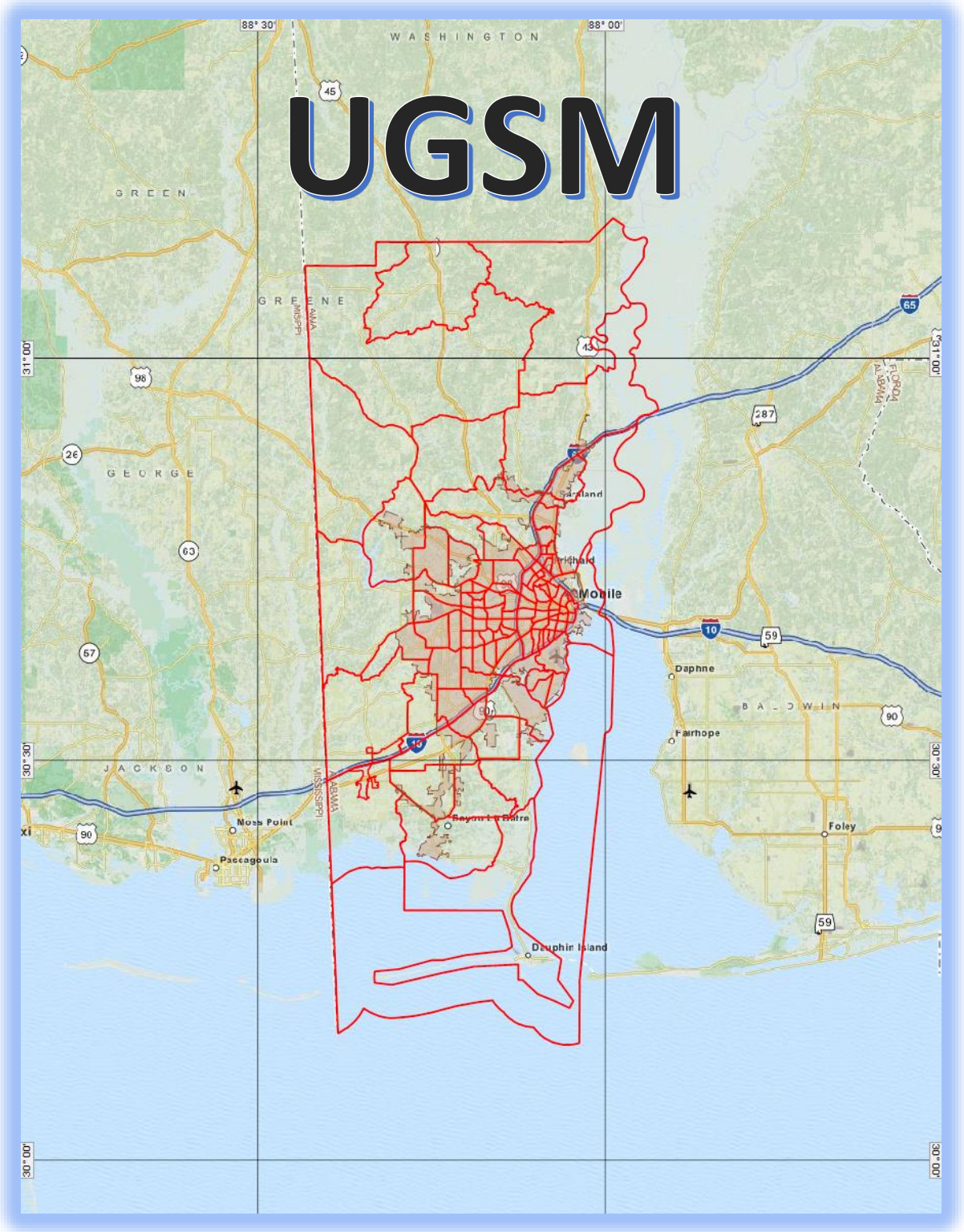
v.3.1 (3/22/2016)



Urbanization growth/shrinkage model

Matthew Kotkowski

“Prediction is very difficult, especially if it's about the future.”

Niels Bohr

Table of Contents

1. Introduction

Many developing countries are experiencing an urbanization growth while their economies develop and people are moving from rural areas into the cities. On the other hand, many well devolved countries are going through an urbanization shrinkage when their population is decreasing for instance due to the low birth rate or economic downfall.

Urbanization Growth/Shrinkage Model (UGSM) described in this paper is one of the many models for predicting urbanization change in the future. The difference between those models are based on the factors they used for prediction as well as on the prediction mechanism.

There are many factors responsible for the urbanization change. The main model from which those particular factors have been taken under consideration to implement in the UGSM model was a SLEUTH Cellular Automata model (SLEUTCH CA). UGSM adopts also some statistical factors like demographics or change in income. The whole purpose of developing a new model, instead of just using the SLEUTH CA model, was to increase the accuracy as well as to decrease the computational expensiveness of the SLEUTH CA model. That is why the UGSM has been devolved. It is by no means author's statement that the UGSM is a finished model. SLEUTH CA model is one of the best CA model and it has been developed through years of tests. In author's opinion the UGSM could prove to be a very successful urbanization change model. As shown later in this paper – the results are very promising and the features selected for the UGSM have proven to be the important factors for predicting the urbanization change.

The paper consists of five major part. In the Methodology – the main concepts of the SLEUTH CA and UGSM are being described and the similarities and differences of those two models. Methodology is being followed by the Experiment, which consists of a few parts describing hardware and software used as well as the description of the implementation and most important code blocs. Lastly, the results are presented and the whole paper is concluded, respectively in Results and Conclusion sections.

2. Methodology

As stated before - the development of the UGSM model has been inspired by the SLEUTH CA model. SLEUTH CA model is an effect of the extreme trial and error technique used to develop its core features. THE UGSM differs in many ways but incorporates into its engine the main SLEUTH CA features. First let's take a look into a SLEUTH CA model to better understand how it works and which features were useful in developing a new model.

2.1 SLEUTH CA model

A cellular automaton (CA) is a model of a system of "cell" objects with the following characteristics.

- The cells live on a ***grid***.
- Each cell has a ***state***. The number of state possibilities is typically finite. The simplest example has the two possibilities of 1 and 0 (otherwise referred to as "on" and "off" or "alive" and "dead").
- Each cell has a ***neighborhood***. This can be defined in any number of ways, but it is typically a list of adjacent cells

off	off	on	off	on	on
on	off	off	off	on	on
on	off	on	on	on	off
off	off	on	off	on	on
on	on	off	off	on	off
on	on	on	off	off	on
on	off	off	on	on	on
off	off	on	off	on	off

Table 1. An example of a grid of cells with values "on" or "off"

The Sleuth CA model has been developed in the Department of Geography, University of California, Santa Barbara (Guan). At first Sleuth was applied to the San Francisco Bay area (Clarke and Gaydos, 1998). Sleuth consists of The Land Cover Deltatron Model (LCD) and Urban Growth Model (UGM) (Chaudhuri and Clarke, 2013). The LCD Model operates in change space. The UGM simulates the growth of urbanization on a specific area with a use of a modified CA. The model has been named after input layers required by the model: **slope, land-use, exclusion, urban, transportation and hill-shad**e (Guan and Clarke, 2010).

Land use change depends on interaction between humans and the environment (Veldkamp and Verburg, 2004). Just like in other CA models – the land is being divided into cells by the use of a grid. The Sleuth model is no different – here the automaton executes transition rules for each particular cell by taking into consideration the neighboring cells of a cell being currently evaluated. Usually one year of urban growth is equal to one growth cycle and a simulation consists of a series of growth cycles (Guan and Clarke, 2010).

Four growth rules are the heart of urban dynamics in the Sleuth model. Those rules are: **spontaneous new growth, new spreading center growth, edge growth and road influenced growth** (Clarke 2008). Those four growth rules are applied in a set of nested loops (Silva, 2002). They depend on five parameters, which are also called coefficients: dispersion, or the likelihood of any pixel turning urban; breed, or the likelihood of cells starting their own growth trajectory; spread, or outward expansion of existing urban areas and infill; slope, or the degree of resistance of urbanization to growing up steep slopes and road gravity, or the attraction of new development towards roads (Clarke, 2008).

2.2. The UGSM model in relation to the SLEUTH CA model

- **The grid, cell and point**

One of the most important element of the SLEUTH CA model is a grid. All operations are being conducted on a grid. The grid is also a foundation of the UGSM. Here the grid is being described in terms of longitude/latitude pairs of points. Those points are centroids of each cell. The size of a cell is known and it can be adjusted, which will be shown in the Implementation section.

- **The transportation network**

In SLEUTH CA a road influenced growth plays an important role in predicting the urbanization change. That is why the transportation network has been added to the UGSM – the distance from each cell represented by the centroid point of each cell to the nearest road. For the Mobile county, AL one transportation layer (in the form of a SHP file) has been added due to the lack of more SHP files so we assume that for each period the transportation network was the same. In the future – it could be useful to add a transportation layer for each time period.

- **Spontaneous new growth, new spreading center growth, edge growth**

By choosing machine learning to perform the prediction – those three growth rules are automatically implemented by adding the following features into a training and testing data:

1) Distance from each cell to the nearest urbanized cell
2) Distance from each cell to Downtown Mobile

- **Slope, land-use, exclusion, urban and hill-shade**

Slope and urban layers has been implemented while hill-shade was not that important for the Mobile county. The exclusion layer can be added in order to exclude specific cells from urbanization (for instance: watershed areas).

3. Experiment

The Linux Ubuntu server has been built and R programing language chosen to conduct this experiment. The specifications of the hardware and software used are described in the following subsections. Because there was no 65-bit version of RStudio (IDE for R) for Windows operating system – it was a good choice to use Linux with 64-bit RStudio. After hardware and software subsections – the implementation part describes the process of obtaining the prediction results.

3.1. Hardware:

Two Xeon E5-2680 CPU units have been installed into a two socket motherboard. Each CPU unit has been equipped with a water cooling system. The 128 GB of DDR3 memory should provide enough working memory in order to perform the computation. Lastly – the reliable and fast SSD drive (up to 550MB/s) to provide enough speed and size for the necessary software.

Motherboard	Asus DDR3 1066 Intel-LGA 2011 Motherboard Z9PA-D8 (ASMB6-IKVM)
CPUs	**2x** Intel Xeon E5 2680 V2 ES 2.8Ghz 25MB 10 Core LGA2011 22nm 130W L2 QE4Z Processor
Memory	**128 GB** of Kingston Technology ValueRAM 1600MHz DDR3 (PC3-12800) ECC Reg CL11 DIMM DR x4 Server Memory KVR16R11D4/16
HDD	Samsung 850 PRO 256GB 2.5-Inch SATA III Internal SSD (MZ-7KE256BW)

Table1 - Characteristics of main hardware components

Picture 1. Ubuntu server performing the computation

3.2. Software:

1. **Ubuntu 14.04 LTS**
2. **R** - free software environment for statistical computing and graphics.
   Version : 3.2.3 54-bit
   Necessary packages:

   a. **library(maptools)**
   b. **library(FedData)**
   c. **library(raster)**
   d. **library(rasterVis)**
   e. **library(rgeos)**
   f. **library(sp)**
   g. **library(dismo)**
   h. **library(mapplots)**
   i. **library(rgdal)**
   j. **library(spdep)**
   k. **library(ggplot2)**
   l. **library(ggmap)**
   m. **library(h2o)**
   n. **library(randomForest)**
   o. **library(geosphere)**
   p. **library(cleangeo)**
   q. **library(grid)**
   r. **library(doParallel)**
   s. **library(foreach)**
   t. **library(fields)**
   u. **library(beepr)**
   v. **library(shapefiles)**

3. **RStudio** – IDE for R
   Version : RStudio 0.99.892 - Ubuntu 12.04+/Debian 8+ (64-bit)
4. **OpenJDK (an open-source implementation of the Java SE Platform)** for h20 package in R
5. **ExpertGPS** – for working with SHP files. The operation of converting the SHP files has been performed on a Windows 10 desktop and SHP files have been saved and open in R environment on Ubuntu server.

3.3. Implementation

1. Creation of a grid

The foundation of this model is a grid, which consists of centroid points representing cells. The **grid** variable creates the grid depending on the size of the Range variable. In order for a prediction mechanism to work is to have sufficient number of data. That is why the Range of a 0.001 seemed sufficient for the purpose of this Experiment. The area of each cell has been calculated and is equal 2651^2m.

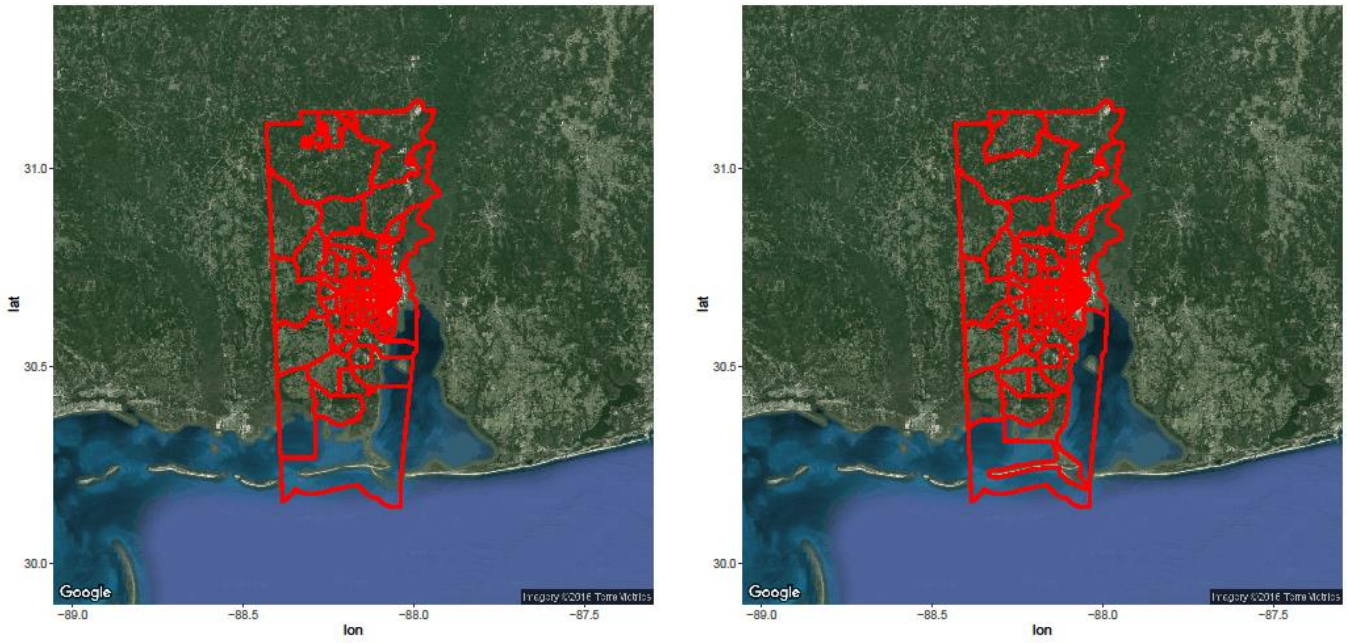
2. Training and testing data

Census2000 (Figure 1a) is used to create a training data set while Census2010 (Figure 2a) creates the testing data set. The whole procedure of creation of those two data sets (training and testing) is similar and requires generating and acquiring following features for each cell:

1) Statistical demographic data from Census2000 and Census2010 SHP files – respectively for a training and testing data

a) Importing Census2000 and Census2010 SHP files

The statistical data on the level of the US Census Tracts has been acquired in a form of a SHP files which consists of polygons representing each Tract region and statistical data for each polygon.

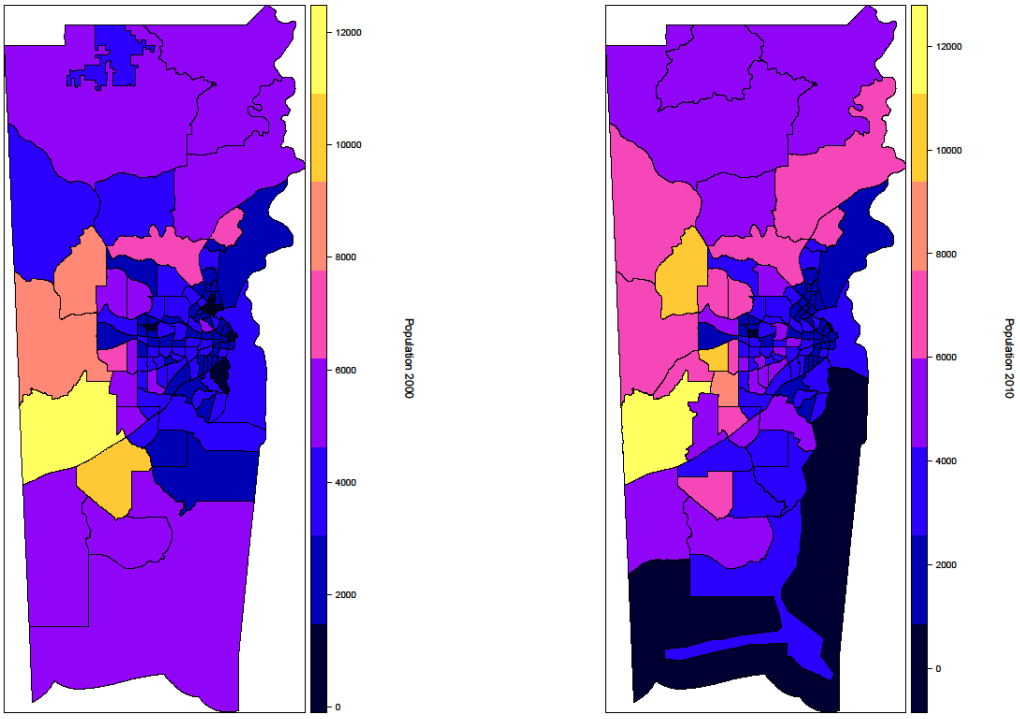


a) 2000 b) 2010

Figure 1 – US Census Tracts

The total population distribution is visible on Figure 2 (Figure 2a for 2000 and Figure 2b for 2010).

Urbanization growth/shrinkage model



b) Importing ZipCode SHP file

It was an assumption that the ZipCoode boundries (Figure 3) have not changed from 2000 to 2010, since author could not find ZipCode SHP files for different periods. The reason of importing ZipCodes SHP file was to include the Median Household Income which was not included in either USCensus2000 or USCensus2010 SHP file.

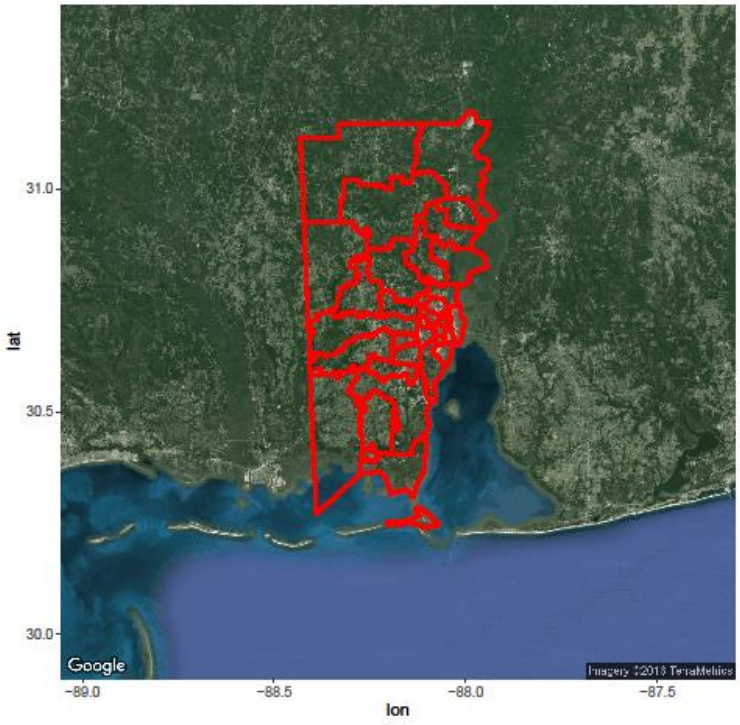


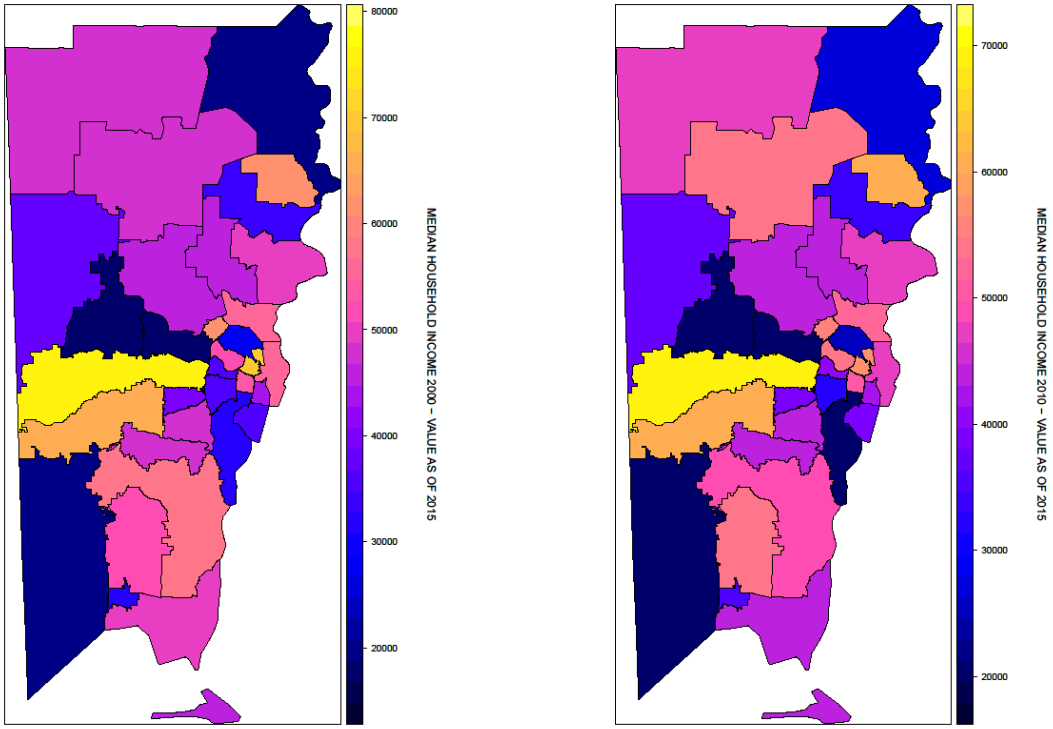
Figure 3. Zip Codes boundaries

2) Median Household Income on the level of a Zip Code

The Median Household Income (MHI) for 2000 (for a train data) and for 2010 (for a test data) loaded from an XLS file for each Zip Code. I am assuming here that the Zip Code boundaries have not changed (Figure 3) – at least I have not been able to find any information that the boundaries have changed.

Since values for 2000 expressed in US dollars have not the same value as MHIs in 2010 – it was necessary to convert MHIs values for 2000 by using a special conversion rates obtained from

Figures 4a and 4b shows respectively the MHI values in 2015 for 2000 and 2010 for each Zip Code.

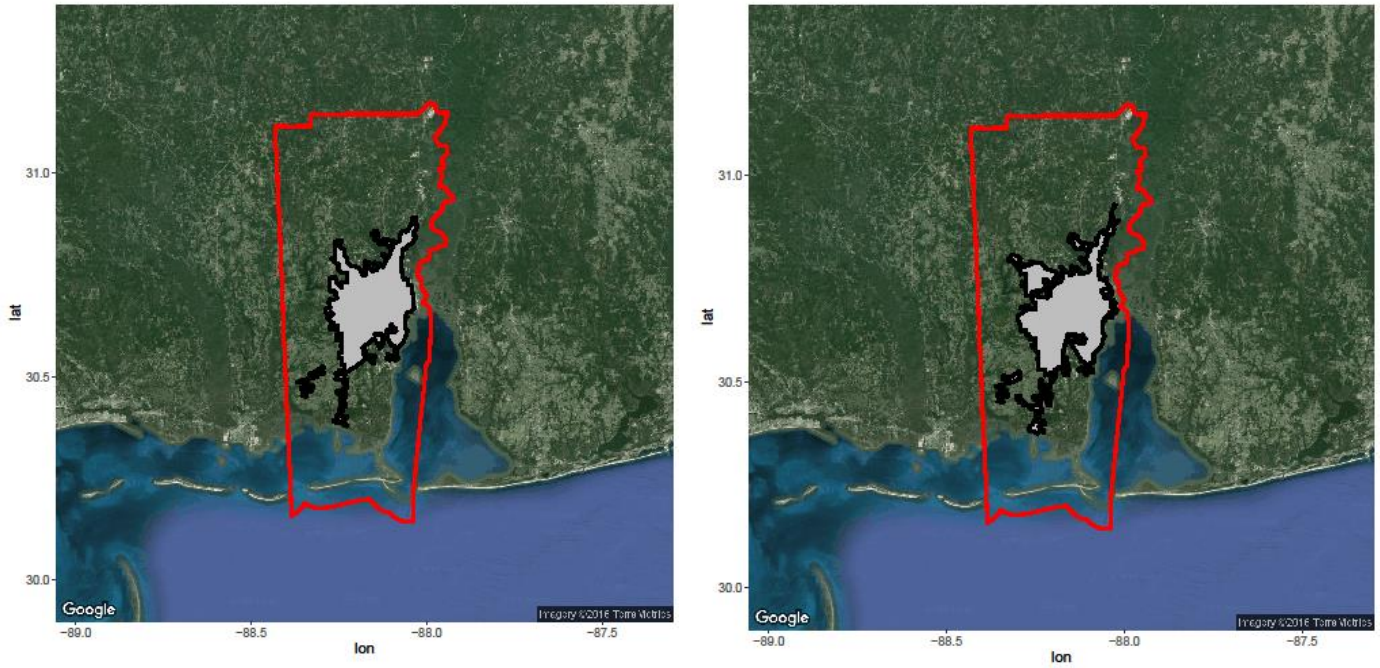


a) MHI value for 2000 b) MHI value for 2010

Figure 4 – MHI values in 2015

3) Urbanization in the previous time period (0 if not-urbanized, 1 if urbanized)

For training data, it is a column of all urbanized cells in 2000 and for training data it is a column of all urbanized cells in 2010.

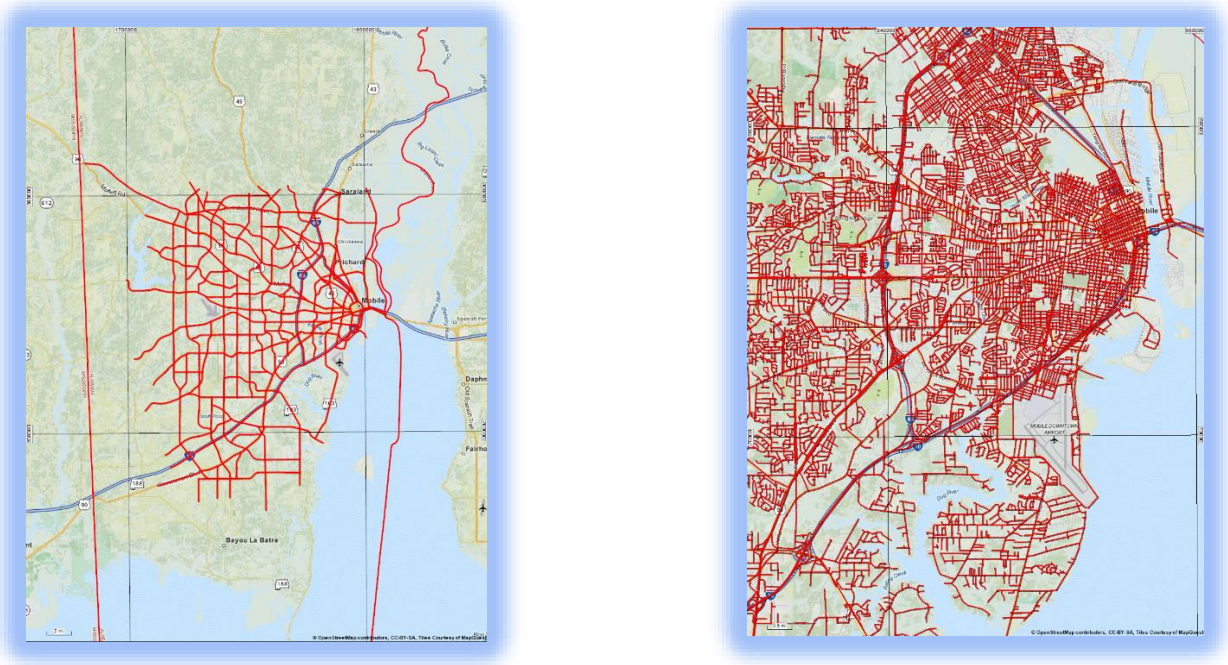


a) 2000 b) 2010

Figure 5 – urbanization areas

4) Distance to the nearest road

For each cell in a grid I am calculating the distance from that cell to the nearest road cell. Right now the major road network has been implemented and is constant for each time period (2000 and 2010) since there were no different SHP files for each period. To increase the computational time, I have developed an algorithm which looks for the distance to the nearest road for each cell in the radius of 0.5 longitude and 0.5 latitude. Such tactic dramatically reduces the computational expense because the irrelevant distant road cells would not be taken under consideration. On Figure 6a implemented major road network is visible and Figure 6b shows a big road network which can easily be implemented to the program but which would require much more computational time to complete the distance calculations (in a new version of the code).



a) Major road network (implemented)

b) Big road network (to be implemented)

Figure 6 – Road Network

5) Distance to the nearest urbanized cell

For each cell in a grid I am calculating the distance from that cell to the nearest urbanized cell. For a training data it is a distance from each cell on the grid to the Urbanized area in 2000 and for testing data it is a distance form each cell to the Urbanized area in 2010.

6) Distance to the Downtown, Mobile

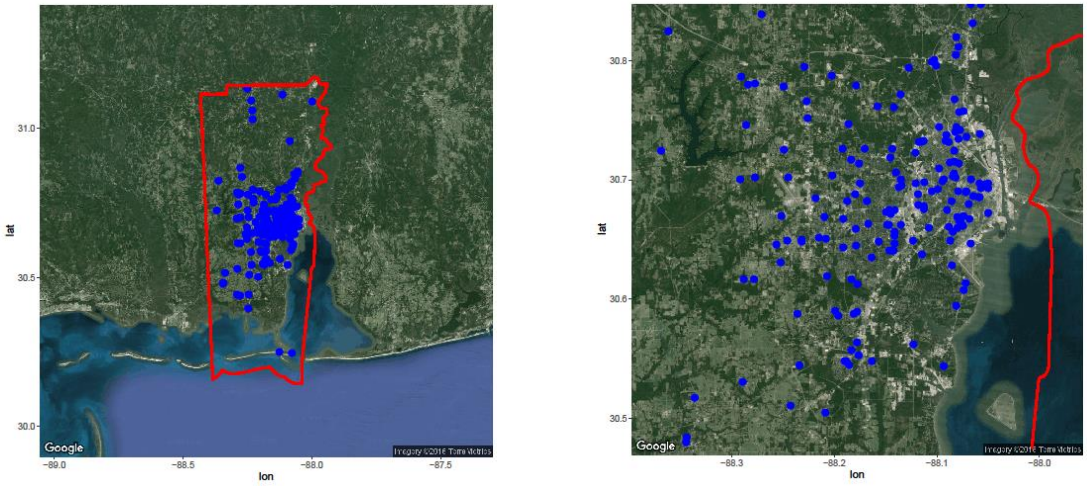
Because the city of Mobile is an urbanization force – I am providing a distance for each cell on a grid to the city center (Downton Mobile, AL). The longitude and latitude of the Mobile city center has been obtained from Mobile's Wikipedia website (Table 2).

Longitude	Latitude
-88.04306	30.69444

Table 2 - Coordinates of the Downtown Mobile center point

7) Distance to the nearest school

The coordinates of each school in Mobile county has been obtained as a shape points file in SHP format. It consists of all kind of schools and in the future schools should be divided into categories (middle, high, colleges etc.). The figure 7 shows the locations of each school in a zoom 9 (Figure 7a) and in a zoom 11 (Figure 7b).



a) zoom 9 b) zoom 11

Figure 7 – Schools in Mobile county

8) Elevation

The elevation was an important factor in the SLEUTH CA and hence it was implemented here. The elevation and slope raster map has been downloaded and the elevation value has been obtained for each cell in the grid. Mobile county, AL is a mostly flat land so the elevation values do not differ much. All the NA values were assumed to be the negative value and have been discarded from the model since the urbanization would not likely happen on the areas under the sea level. The Figure 8 is a plot of the elevation raster map for the entire United States. Such a map could be easily used for any other area in the United States.

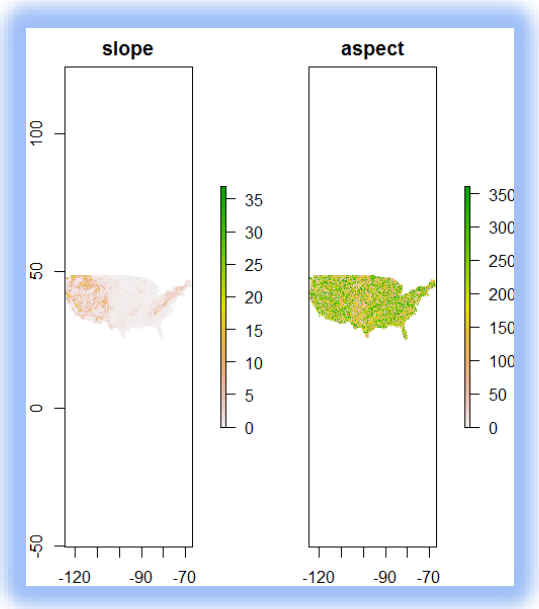


Figure 8 - Elevation and the slope from an USA raster map

1. Prediction

Different prediction mechanism has been used to find features importance than to make a final prediction because randomForest features importance function is much more reliable and H2o deep learning mechanism should give on other hand more accurate predictions since it is utilizing the neural network prediction mechanism. H2o is also very handy since it allows to set the number of CPU threads to be used as well as maximum amount of memory. In this way the Ubuntu server was able to use all of its 40 threads and about 120 GB of RAM to look for the best set of parameters and obtain the final model and make a prediction using that final model. Thus the necessary steps to obtain the urbanization change values were as follows:

2. Feature Importance

The randomForest has been used to provide the Feature Importance results and a plot. As expected (Figure 9) – the most important features are:

1) x.downtown_distance
2) x.elevation
3) x.deltaIncome
4) x.distance_to_urban
5) x.distance_road
6) x_distance_school

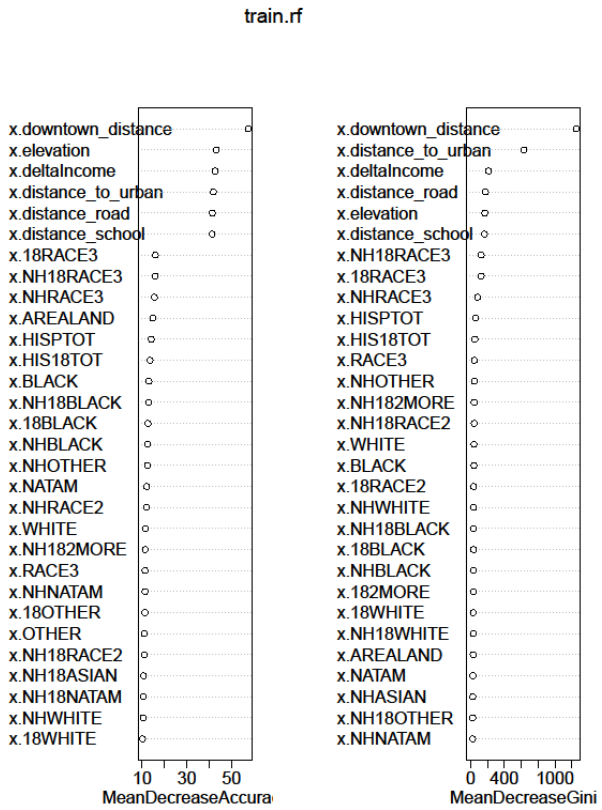


Figure 9 - Random Feature importance

Obtaining the best set of parameters for the final prediction

The hyper parameter search from H2o package has been used to check the possible sets of parameters in order to maximize the AUC metric (Area Under Curve). This metrics is used for a classification prediction in order to determine the percentage of properly predicted categories. In this case there are two categories: 1 and 0. One means that the cell will be urban and zero means that the particular cell will not be urban.
In order to minimalize the computational expense, the train data set has been divided in half and from the first half randomly has been selected 20% cells which was around 20,000 of cells. Also from a train set in the same way (but from the other half of a train set) – the test set has been created which was close to the 20,000 of cells as well. The train set was a data frame for H2o hyper parameter search and the test set was a validation set. In this way the AUC metrics could be computed. As many as 48 models has been checked (Table 3). and the best one has been selected to perform the final prediction (Table 4).

model_id	model_hidden1	model_hidden2	model_hidden3	model_l1	model_epochs	activation	auc
Grid_DeepLearning_train_model_R_1458268249925_4_model_3	1024	512	256	1.00E-07	62.91378	Rectifier	0.921741
Grid_DeepLearning_train_model_R_1458268249925_4_model_10	100	300	100	1.00E-07	54.87481	Rectifier	0.93192
Grid_DeepLearning_train_model_R_1458268249925_4_model_32	100	300	100	1.00E-05	54.87481	RectifierWithDropout	0.946599
Grid_DeepLearning_train_model_R_1458268249925_4_model_18	500	500	500	1.00E-07	52.84614	Tanh	0.951877
Grid_DeepLearning_train_model_R_1458268249925_4_model_19	500	500	500	1.00E-07	60.1114	Tanh	0.937059
Grid_DeepLearning_train_model_R_1458268249925_4_model_16	500	500	500	1.00E-05	52.24101	Tanh	0.938601
Grid_DeepLearning_train_model_R_1458268249925_4_model_34	100	300	100	1.00E-07	54.87481	RectifierWithDropout	0.94549
Grid_DeepLearning_train_model_R_1458268249925_4_model_20	100	300	100	1.00E-05	54.87481	Tanh	0.949112
Grid_DeepLearning_train_model_R_1458268249925_4_model_14	1024	512	256	1.00E-07	49.36106	Tanh	0.936974
Grid_DeepLearning_train_model_R_1458268249925_4_model_27	1024	512	256	1.00E-07	64.11065	RectifierWithDropout	0.948224
Grid_DeepLearning_train_model_R_1458268249925_4_model_15	1024	512	256	1.00E-07	54.9064	Tanh	0.937944
Grid_DeepLearning_train_model_R_1458268249925_4_model_35	100	300	100	1.00E-07	64.11065	RectifierWithDropout	0.947804
Grid_DeepLearning_train_model_R_1458268249925_4_model_45	100	300	100	1.00E-05	64.11065	TanhWithDropout	0.930263
Grid_DeepLearning_train_model_R_1458268249925_4_model_22	100	300	100	1.00E-07	54.87481	Tanh	0.935311
Grid_DeepLearning_train_model_R_1458268249925_4_model_6	500	500	500	1.00E-07	54.87481	Rectifier	0.92013
Grid_DeepLearning_train_model_R_1458268249925_4_model_38	1024	512	256	1.00E-07	47.73889	TanhWithDropout	0.886128
Grid_DeepLearning_train_model_R_1458268249925_4_model_43	500	500	500	1.00E-07	60.24537	TanhWithDropout	0.929364
Grid_DeepLearning_train_model_R_1458268249925_4_model_36	1024	512	256	1.00E-05	51.24768	TanhWithDropout	0.908239
Grid_DeepLearning_train_model_R_1458268249925_4_model_41	500	500	500	1.00E-05	51.44545	TanhWithDropout	0.945384
Grid_DeepLearning_train_model_R_1458268249925_4_model_39	1024	512	256	1.00E-07	56.01378	TanhWithDropout	0.907544
Grid_DeepLearning_train_model_R_1458268249925_4_model_28	500	500	500	1.00E-05	54.87481	RectifierWithDropout	0.93716
Grid_DeepLearning_train_model_R_1458268249925_4_model_47	100	300	100	1.00E-07	64.11065	TanhWithDropout	0.920889

Grid_DeepLearning_train_model_R_1458268249925_4_model_0	1024	512	256	1.00E-05	54.87481	Rectifier	0.939185
Grid_DeepLearning_train_model_R_1458268249925_4_model_17	500	500	500	1.00E-05	58.90835	Tanh	0.941592
Grid_DeepLearning_train_model_R_1458268249925_4_model_9	100	300	100	1.00E-05	64.11065	Rectifier	0.924643
Grid_DeepLearning_train_model_R_1458268249925_4_model_2	1024	512	256	1.00E-07	54.87481	Rectifier	0.924374
Grid_DeepLearning_train_model_R_1458268249925_4_model_11	100	300	100	1.00E-07	64.11065	Rectifier	0.926924
Grid_DeepLearning_train_model_R_1458268249925_4_model_40	500	500	500	1.00E-05	37.14021	TanhWithDropout	0.945005
Grid_DeepLearning_train_model_R_1458268249925_4_model_1	1024	512	256	1.00E-05	61.7144	Rectifier	0.940911
Grid_DeepLearning_train_model_R_1458268249925_4_model_23	100	300	100	1.00E-07	63.51127	Tanh	0.945625
Grid_DeepLearning_train_model_R_1458268249925_4_model_42	500	500	500	1.00E-07	51.50111	TanhWithDropout	0.915447
Grid_DeepLearning_train_model_R_1458268249925_4_model_13	1024	512	256	1.00E-05	57.03148	Tanh	0.950195
Grid_DeepLearning_train_model_R_1458268249925_4_model_25	1024	512	256	1.00E-05	63.49876	RectifierWithDropout	0.951756
Grid_DeepLearning_train_model_R_1458268249925_4_model_5	500	500	500	1.00E-05	61.11752	Rectifier	0.938453
Grid_DeepLearning_train_model_R_1458268249925_4_model_29	500	500	500	1.00E-05	64.11065	RectifierWithDropout	0.949369
Grid_DeepLearning_train_model_R_1458268249925_4_model_31	500	500	500	1.00E-07	64.11065	RectifierWithDropout	0.933635
Grid_DeepLearning_train_model_R_1458268249925_4_model_37	1024	512	256	1.00E-05	46.78183	TanhWithDropout	0.914058
Grid_DeepLearning_train_model_R_1458268249925_4_model_7	500	500	500	1.00E-07	61.11689	Rectifier	0.931645
Grid_DeepLearning_train_model_R_1458268249925_4_model_4	500	500	500	1.00E-05	54.87481	Rectifier	0.933502
Grid_DeepLearning_train_model_R_1458268249925_4_model_44	100	300	100	1.00E-05	54.87481	TanhWithDropout	0.919329
Grid_DeepLearning_train_model_R_1458268249925_4_model_12	1024	512	256	1.00E-05	50.36749	Tanh	0.950432
Grid_DeepLearning_train_model_R_1458268249925_4_model_21	100	300	100	1.00E-05	62.91378	Tanh	0.941891
Grid_DeepLearning_train_model_R_1458268249925_4_model_8	100	300	100	1.00E-05	54.87481	Rectifier	0.922712
Grid_DeepLearning_train_model_R_1458268249925_4_model_30	500	500	500	1.00E-07	54.87481	RectifierWithDropout	0.94536
Grid_DeepLearning_train_model_R_1458268249925_4_model_26	1024	512	256	1.00E-07	54.87481	RectifierWithDropout	0.936709
Grid_DeepLearning_train_model_R_1458268249925_4_model_33	100	300	100	1.00E-05	64.11065	RectifierWithDropout	0.949349
Grid_DeepLearning_train_model_R_1458268249925_4_model_46	100	300	100	1.00E-07	54.87481	TanhWithDropout	0.922752
Grid_DeepLearning_train_model_R_1458268249925_4_model_24	1024	512	256	1.00E-05	54.87481	RectifierWithDropout	0.944878

Table 3 – List of models

V1					
Grid_DeepLearning_train_model_R_1458268249925_4_model_18					
500					
500					
500					
1.00E-07					

Urbanization growth/shrinkage model

52.84614					
Tanh					
0.951877					

Table 4 – Best model

For the final model – the best model's parameters have been chosen and whole train data set and for a prediction the final model and whole test set.

4. The Results

The resulting data frame consisted of the urbanization area in a shape of cells for each set of longitude and latitude for 2020. After subtracting the cells which have not changed and plotting the points which have changed on the map in two colors – the resulting plot is a force change plot. The blue color represents the positive force of urbanization and the red color represents the negative force of cells becoming un-urbanized. The grey area is a non-affected zone which is neither more urbanizing on un-urbanizing (neutral zone). The urbanization force is visible on figure 7.

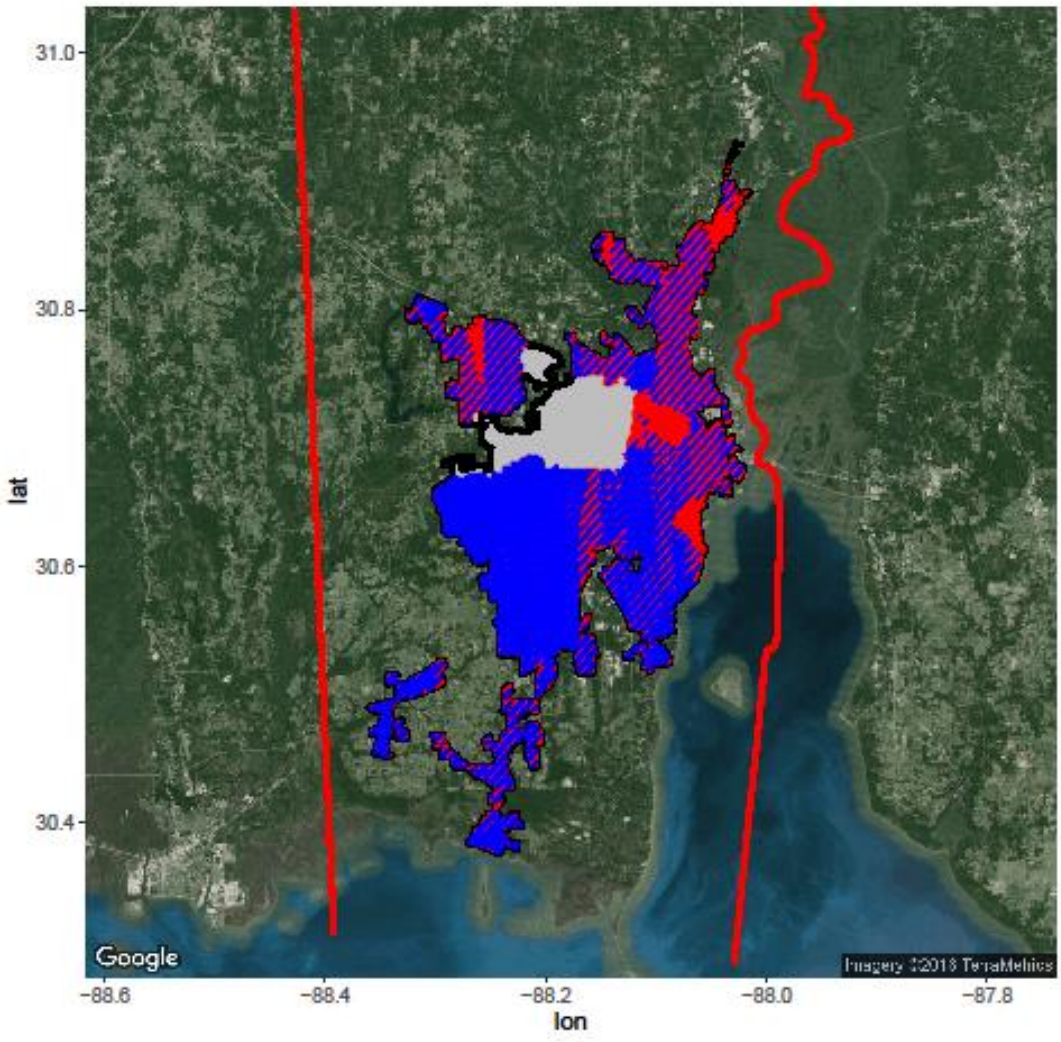


Figure 10 – Final Prediction, urbanization force

The next figure (Figure 11) shows areas most likely to become urbanized. In comparison to the above figure – instead of showing a positive or negative urbanization force – it simply shows which areas outside urbanization in 2010 would possibly become urbanized in 2020. The areas which could lose the urbanization status can be seen in red color in the figure 10.

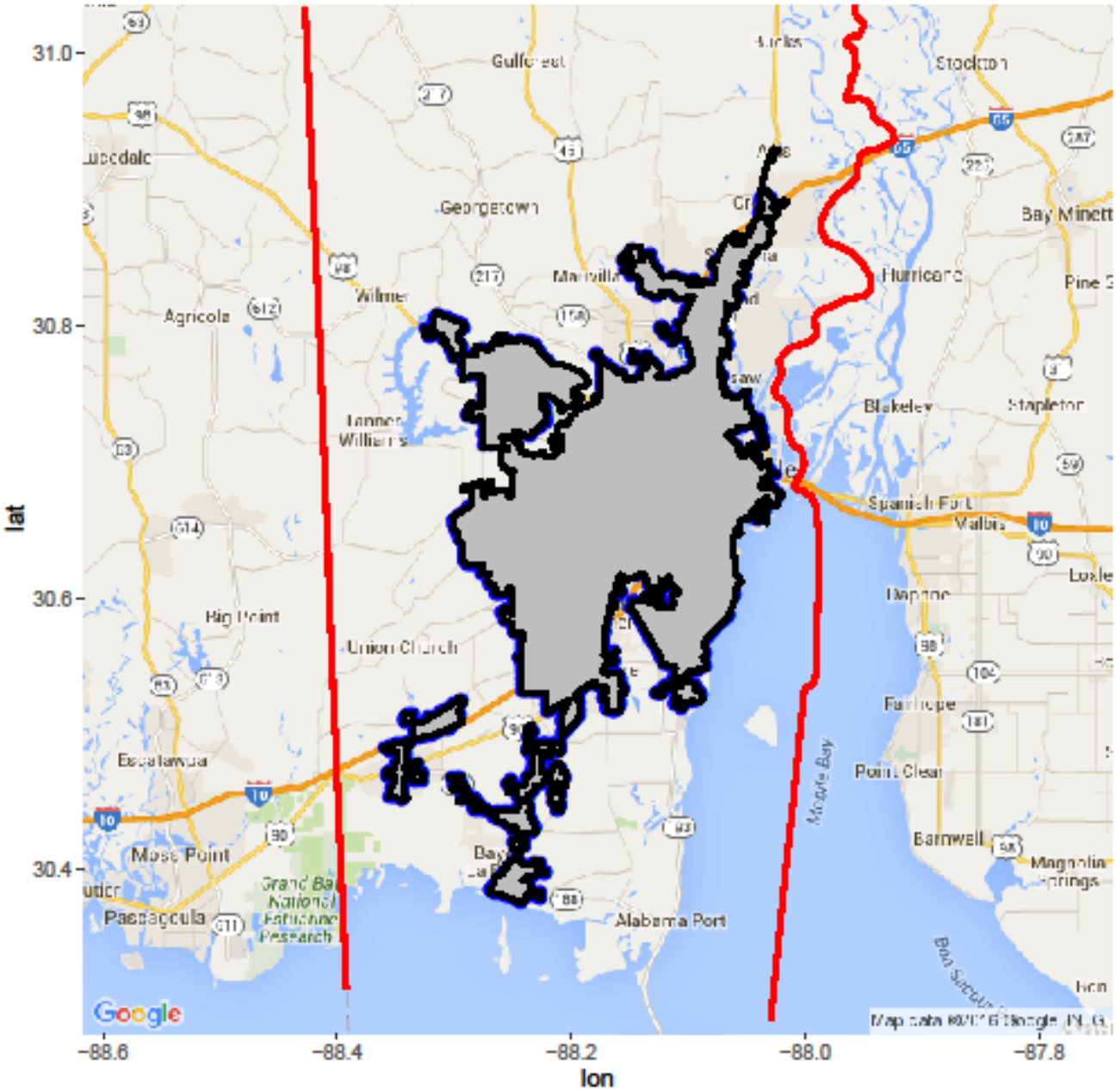


Figure 11 – Final Prediction, newly urbanized cells

5. Further Work:

1. **Road-network**
   Instead of a general road network – implementation of a more detailed road-network (need to update a finding minimal distance algorithm to minimalize the computational expense).
   The idea is to look for a minimal distance between each cell and roads within 50 miles instead of a looking for a distance between each cell and all the roads represented by polylines. This way it will decrease the computational time.

2. **Distance to the nearest water area (water cell)**
   Everything is ready to calculate the distance from each cell on the grid to the nearest water area – it just requires to obtain a reliable SHP file. Unfortunately, all SHP files containing water areas had flows which were disqualifying them. The water area for the Mobile county could improve the accuracy of the model since it is a county close to the Bay of Mexico and some urbanized could be a result of vocational destination of many Americans.

3. **Distance to the nearest shopping center**
   It is possible to add shopping centers just like the schools were added and to calculate the distance from each cell to the nearest major shopping center.

4. **Crime index**
   For each cell in the grid a violent crime index could be obtained either on the level of Zip Codes or lower level like US Census Tracts or even US Census Block Groups or US Census Blocks

5. **Features selection**
   The data acquired from a Mobile county GIS website encapsulates the typical demographics data divided into a little more than 60 categories, while US Census data for 2000 and 2010 has more than 8000 categories to chose from. Because of the limitations of my server – it was necessary to build a model with limited number of features. That is why the SHP files provided by the Mobile county GIS website has been applied to the model.
   In the future it would be very interesting to find more suitable mix of features which could increase the accuracy of the model.
   The median household income is on the ZipCode level while other data is on the Tract level. For the 36615 ZipCode there was no Income data available so it was necessary to get the median income.
   Again, due to the technical limitations – the better accuracy could be reached by importing the data on the level of US Census Blocks Groups or even US Census Blocks. Thus, US Census Blocks level is the lowest statistical level.

6. **Features subdivision**
   Some categories of data could be subdivided further. For instance, the schools could be divided into elementary, high-schools, universities etc. After such subdivision the model should be tested how each subdivision impact the final result.
   Also more parameters could be added like the overall rank of the school in the county. Unfortunately, such ranks of the schools are many times arbitrary and not all schools are graded.

7. **Better algorithms**
   The idea was to limit the usage of the loops but it was not always possible. Some algorithms might be still updated in order to minimalize the loops even further. However, the author was trying his best to create the fastest code possible, it is always possible to improve. Other prediction algorithms should be testes (GMB, randomForest etc.) and the best one chosen for the final model.

8. **Urbanization detection app**

   The model has been built based on the data freely available. The most important issue was to minimalize the computational time since such a models are very computational expensive.
   It could prove useful to detect urbanized areas on Google maps in order to increase the accuracy of the model. Right now – the model does not take under consideration the smaller urbanized area – just the Mobile county metropolitan area available from Mobile county website (for 1990 and 2000) as well as from a US Census website (for 2010).

9. **Localization**

   The model gives the general direction of the urbanization growth/shrinkage. It could be useful to divide the data into smaller areas and run a model for each of them. The desired level would be 30m which is consistent with a SLEUTH CA model.

10. **Shorter predictions**

    The model is based on the U.S. decimal data from 1990, 2000 and 2010. If SHP files representing urbanization areas as well as statistical data would be available for a shorter period (5 years or 1 year) it could potentially increase the accuracy of the prediction if such data would be loaded into the model. The availability of the data as well as the comparability of it (so the features for each period would be similar) is unfortunately not always the case.

11. **Python and C++**

    A nice GUI (Graphical User Interface) should be design where would be possible to upload SHP files for each county, choose the cell size and start the prediction.
    different prediction models could be chosen.

12. **Perfecting the model**

    The SLEUTH CA model – one of the most successful CA model has been improved over the years – yet sometimes it is difficult to say if it is successful in each particular case or not.
    This model (UGSM) is not an exception – in order to properly evaluate it – it should be applied, improved and tested again.
    Constantly testing and improving the code measuring the system time for each independent part of the code. Also there are parts of code that can be simplified and the functions can be used more often instead of repeating some blocks.

6. Conclusion

Every urbanization change model can be fully tested when comparing the prediction to the real-life results. The UGSM model prediction period could be shortened into 5, 2 or even 1-year period instead of a decimal period. The only limit is to acquire reliable data for the prediction mechanism since the data acquisition was the most difficult task. The optimal data should be based on a level of a U.S. Census Block level both for demographic data as well as for economic data. Unfortunately, the economic data was not available at that level and US Census Block SHP files had many errors and were inconsistent. That is why it was necessary to use US Census Tract level for demographic data and US Zip Code level for economic data.

By implementing the exclusion layer into the model – the decision making authorities can use this model to check how their decision could affect the direction of urbanization growth or shrinkage.

To conclude - however, there are still many ways to further improve this model, it seems already to be quiet effective by giving the 95% of AUC on the training data. Such score should be enough to move this model into a next phase of testing and optimizing since every promising model should go through many trial and error phases.

7. The Code:

“The only relevant test of the validity of a hypothesis is comparison of prediction with experience.”

Milton Friedman