Case Study 4 (CS4): Non-Linear Crime Models

"On my honor, I pledge that I have neither given nor received help on this assignment."

Name: _____Yifeng Song_____ Date: _____10/25/2015_____

Introduction

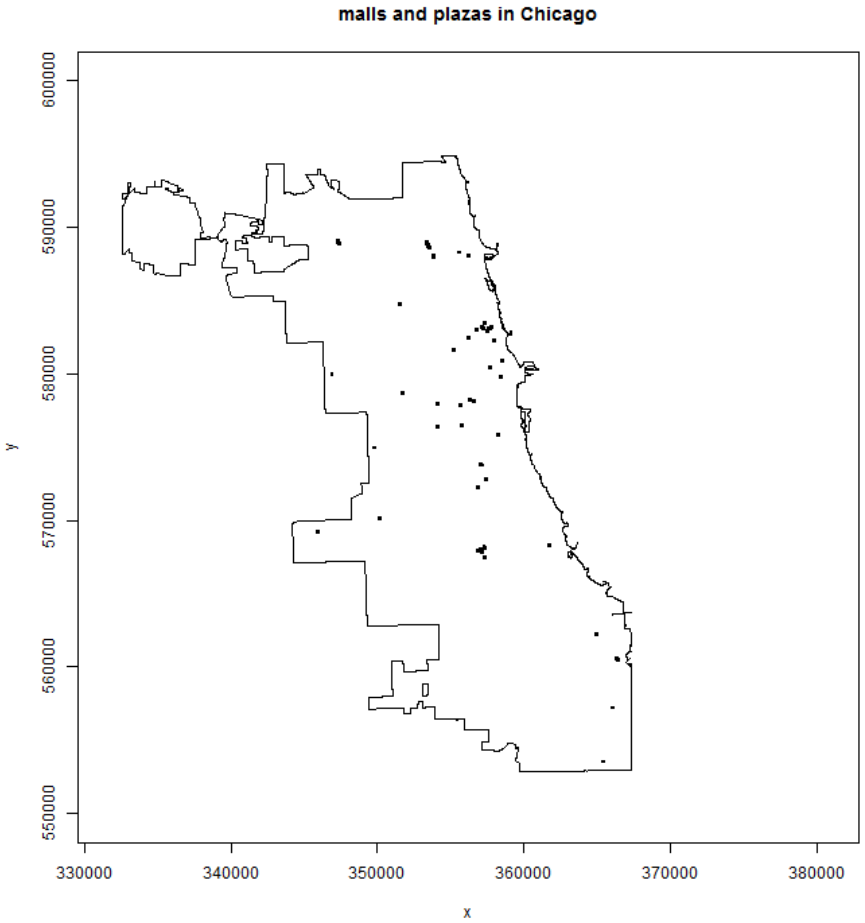
As a continuation of Case Study 3, the case study focuses on investigating the theft crime density in Chicago using a non-linear model called support vector machine (SVM). The Case Study 3 picked 3 spatial factors (distance to the nearest school, police station and park, respectively) and assumed that a linear crime model (logistic regression) can account for these factors on top of kernel density estimation (KDE), so as to make more accurate predictions for the crime density distribution. The results have shown the improvement in the prediction accuracy (by calculating the AUC score in the surveillance plot), but the improvement is very small. So the linear crime model is not sufficient and accurate enough for fully interpreting the spatial factors that affect the crime rates. One possibility is that there are some non-linear effects within those spatial factors which cause the linear model prediction to deviate from the true distribution. Among the 3 spatial factors that were studied last time, distance to the nearest school turned out to improve the prediction accuracy more if it is included in the logistic regression model, while the other 2 factors, especially the distance to the nearest park, seemed to not help improve the prediction accuracy of the logistic regression model. Therefore, in this study where we build the SVM model, these 2 factors (distance to the nearest police stations and parks) are chosen to be the predictors of the classification model.

Another spatial factor that I am interested in studying in the case study is the distance to the nearest shopping center or mall. As I discussed in the first section of Case Study 3 report, there is a much higher crowd density within and near the shopping malls than other places, therefore these areas are prone to becoming the targets of the thefts as they will have a higher chance of success in these densely crowded places. But the dependence of the crime rates on the nearest distance to the shopping center might not be linear, since many shopping centers might take some measures (such as resorting to the police's help, installing video surveillance systems) which can effectively prevent the thefts from occurring, otherwise they will lose a lot of customers as a result of the high crime occurrence rate. Moreover, if two possible spatial factors are very close to each other, their effects on the crime rates may not be simply a linear adding of the two factors together, as there can be some mutual and interacting effects that can cause the crime rates to deviate from the linear dependence. Also, there are plenty of other spatial factors that are not considered in the previous case study, all of which could tweak the linear model prediction. Therefore, the logistic regression model is likely to be a too simplistic crime model, while it is worthwhile to consider the more flexible non-linear crime models such as SVM.

Data collection, Methods, Analysis, Evaluation

This study still uses the 2014 theft crime data of Chicago, which is a CSV file that could be read in R and split into 12 parts according to each month. The shapefile for the malls & plazas of

Chicago is available for downloading on the webpage https://data.cityofchicago.org. Similar to the shapefile for the parks of Chicago (discussed in Case Study 3 Report, the malls & plazas shapefile has the type of “polygon”, with each polygon corresponding to the location of each mall (or plaza). Using the Center_Points() function that was written in case study 3, the estimated coordinate ((x, y) values) of the center point of each mall (or plaza) is calculated and stored. The following picture is the locations of all malls and plazas within the city of Chicago. For the locations of the police stations and parks within the Chicago city boundary, see Case Study 3 report.



Same as the last Case Study, using the get.grid.points() function, an evenly distributed points (resolution = 200 m) within the city boundary were generated, which are considered as the non-crime points in each training set. These points are also where the predictions for the crime probability (density) are made using the SVM models. Then for each month selected as the training part, the location ((x,y) values) of each crime incidence were stored, and they are the monthly crime points. After they are combined with all of the non-crime points, a full training set corresponding to this particular month is generated.

In each training set, all crime points have the response value of 1, while all non-crime points have the response value of 0. To properly build the SVM models using the training set, the response variable needs to be converted into the factor data type in R. There are a total of 4

predictors in the SVM models: x1 is this month's theft density at each training set point estimated by KDE using the previous month's crime locations. x2 is the distance to the nearest police station of each point, x3 is the distance to the nearest park of each point, and x4 is the distance to the nearest mall (or plaza) of each point. x2, x3 and x4 can all be calculated using the get.min.distances() function.

Due to low speed of running SVM models in R, only 3 training/testing set pairs are selected (3 cross validations for each SVM models): February theft data is chosen as the training set, which is used to predict the March theft data; And May theft data is chosen as the training set, which is used to predict the June theft data; and August theft data is chosen as the training set, which is then used to predict the September theft data. All testing sets use the same prediction points, which are just the evenly distributed grid points within the city boundary whose resolution is 200 meters. The x1 value for each prediction (testing) point is calculated based on previous month's (its corresponding training set) actual crime points using the KDE method.
Before starting running the SVM modeling, first the linear crime model (logistic regression) including x1, x2, x3, x4 as predictors are performed for the 3 training sets: February, May, and August. The following is the hypothesis tests for the coefficients of x2, x3, x4 resulting from logistic regression:

	p_x2	**p_x3**	**p_X4**
1	6.65E-13	3.80E-04	0.3601682
2	7.31E-15	1.72E-03	0.7651688
3	1.89E-15	7.60E-05	0.8960373

The p-values of the coefficient of x4 (distance to the nearest mall or plaza) are much bigger than 0.05 in all of the 3 logistic regressions, providing some support for the hypothesis of this case study that there might be some non-linear effects with this predictor as linear crime model does not describe it very well.

Three different SVM models are built to predict the theft occurrence in this study. The first one is the linear SVM model, which seeks to find a flat hyperplane which classifies the binary responses ("0" or "1", non-crime or crime points in this case) with a maximized soft margin. The second SVM model is the degree-2 polynomial kernel SVM, which uses the quadratic kernels to establish the non-linear classification boundary in the feature space. And the third SVM model is the radial basis function (RBF) kernel SVM, which uses the kernels that have the form of exponential functions to establish the non-linear classification boundary. All of these 3 models have a tuning parameter called "C" (cost of constraints violation parameter) in the ksvm() function in R which is used to fit the training data using SVM. The bigger the C value, the "softer" the margin is and the more flexible the SVM model is in terms of allowing for the observations to violate the margin. Ideally, a number of different "C" values will be tested on each model to predict the crime density, and "C" value resulting in the best accuracy will be picked up as the optimal "C" value. However, due to the relatively low running speed of SVM in R and the time constraints of this case study, only a few different "C" values are chosen to run the SVM models.

First, I ran all of the 3 SVM models to fit all 3 training sets with the “C” value being 100 in all cases. Then for linear SVM model, I also tried “C” = 0.01, 0.1, 1, 10 for all 3 training sets. And for the degree-2 polynomial kernel SVM model, I only tried “C”=0.01 and 1 for all 3 training set. Likewise, for the RBF kernel SVM model, I only tried “C”=0.01 and 1 for all 3 training sets.

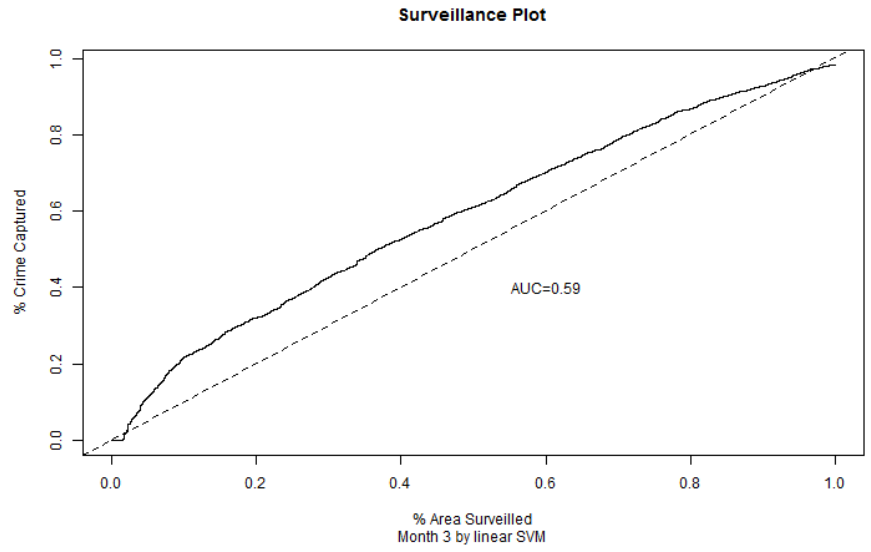
When fitting the training data using the ksvm() function, the argument “prob.model” was set to TRUE so that it will employ a probabilistic SVM model. That is, when predicting the response values in the testing sets, the prediction() function will predict the probability for non-crime (“0”) and crime (“1”) for each testing point, instead of a binary value of “0” or “1”. The sum of the probability of “0” and the probability of “1” is always equal to 1. And in the prediction() function, the argument “type” also needs to be set to “probabilities”. A probabilistic SVM model makes more sense in the surveillance plot – the entire city area is divided into 200 m × 200 m square areas, whose center points are just the evenly distributed grid points within the city boundary. So each area has a predicted probability (or density) of crime associated with it. For the plot.surveillance.curve() function that was used in Case Study 2 and 3, the function first sort all of the squares according to their probability (or density) of crime, in a descending order. Then starting from the highest square area, the number of actual theft incidences in the testing month in each square area are counted, so that for each value of % area surveilled (x-axis), the corresponding % crimes captured (y-axis) can be calculated. Therefore, just like plotting the surveillance curve using the predictions of KDE and logistic regression in the previous case studies, it is more reasonable to use the predicted probabilities for crime (response value “1”) based on SVM models in the surveillance plot step, than to use the predicted binary response values which can only take on the value of either 1 or 0.

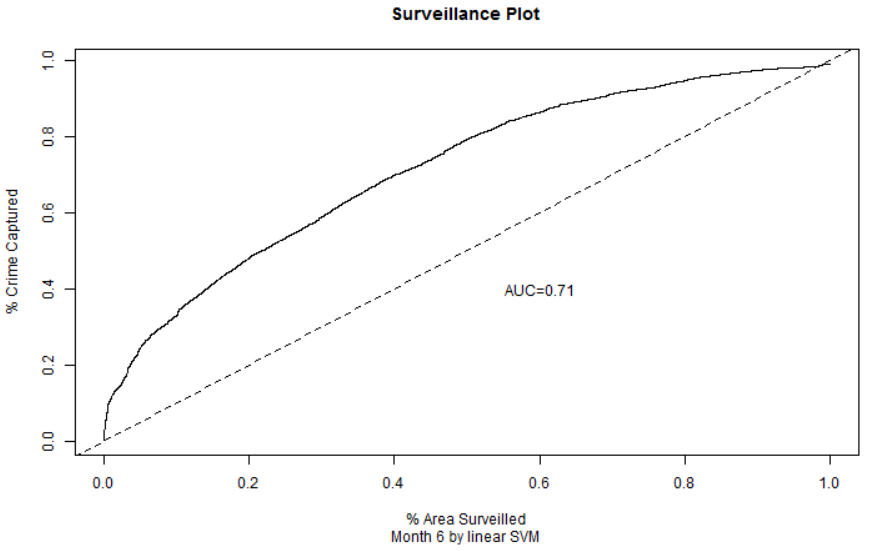
For each unique combination of the SVM model, the “C” value as well as the training month that is used, a surveillance plot is generated and the AUC (Area Under Curve) score is calculated. Same as the logistic regression, AUC score is used to evaluate the performance of the different SVM models. The higher the AUC score, the higher the overall accuracy of the predictions made by the model is. The AUC scores from these non-linear crime models can also be compared with the AUC scores from the linear crime models. Moreover, for a certain type of SVM model (e.g. the linear SVM), the AUC corresponding to different “C” values can be calculated, in order to find out the optimal “C” value for this particular model.

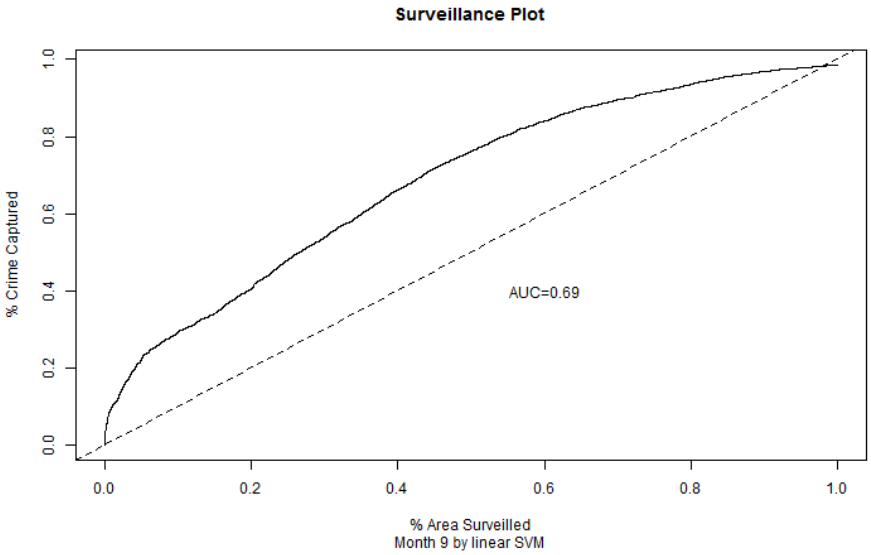
Results and Discussions

The following are the surveillance plots generated from evaluating the different SVM models for each training and testing set pair.

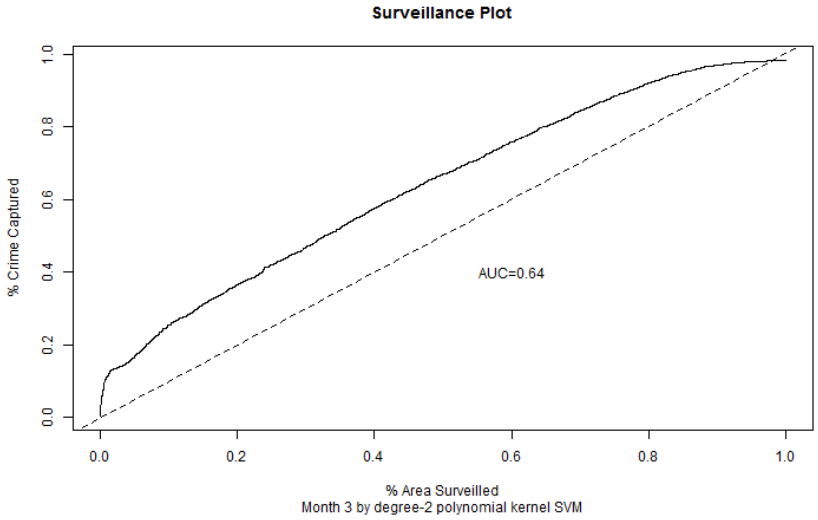
1. Linear SVM (“C”=100, for surveillance plots of other “C” values, see the Appendix):

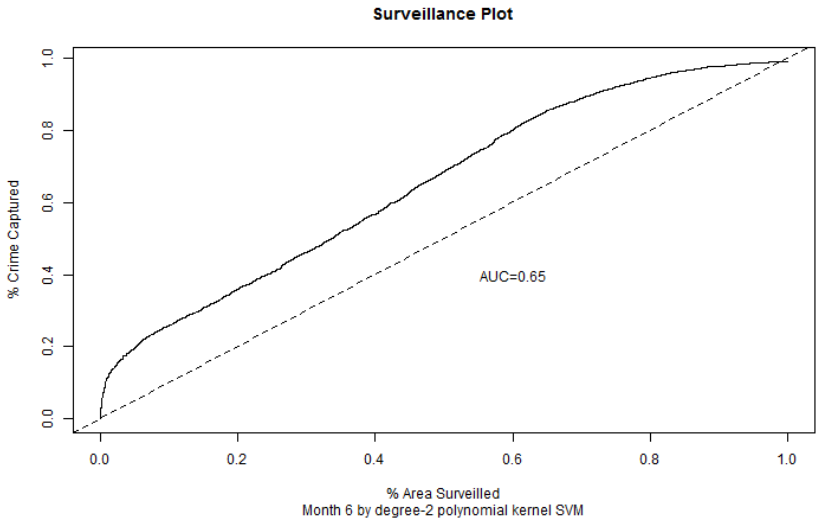

Surveillance Plot
% Crime Captured
0.0
0.2
0.4
0.6
0.8
1.0
AUC=0.59
% Area Surveilled
Month 3 by linear SVM

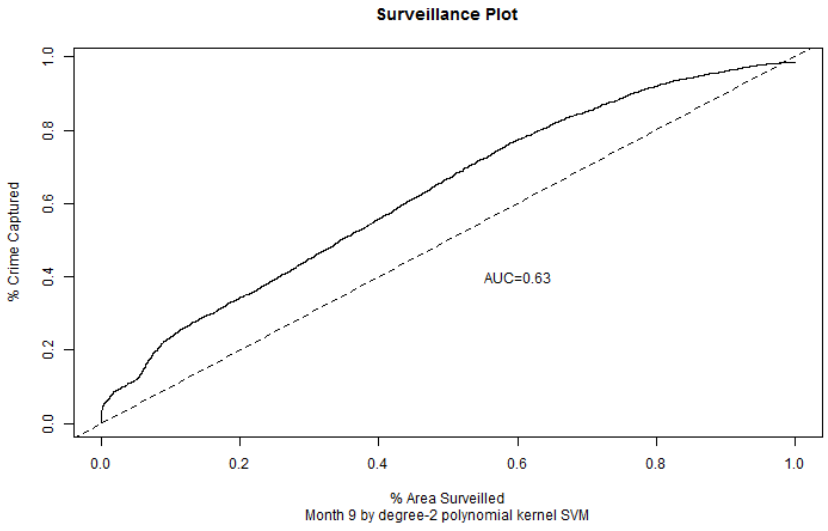

Surveillance Plot
% Crime Captured
0.0
0.2
0.4
0.6
0.8
1.0
AUC=0.71
% Area Surveilled
Month 6 by linear SVM


Surveillance Plot
% Crime Captured
0.0
0.2
0.4
0.6
0.8
1.0
AUC=0.69
% Area Surveilled
Month 9 by linear SVM

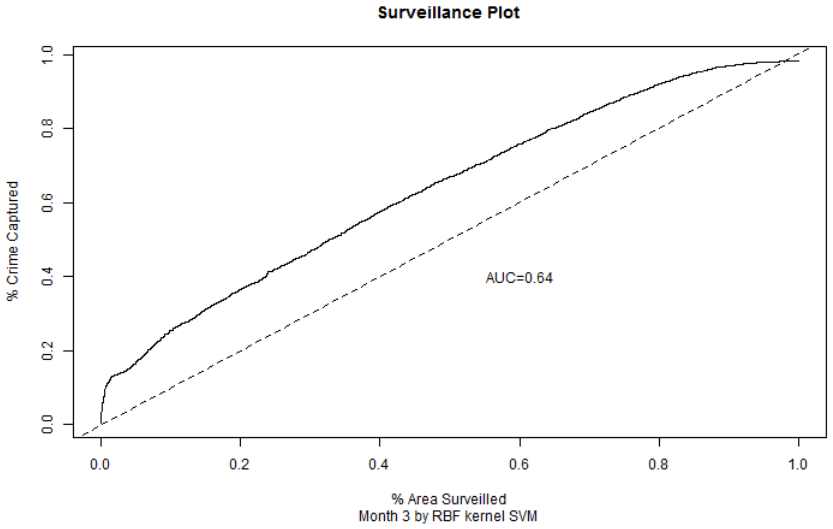
2. Degree-2 polynomial kernel SVM (“C”=100, for surveillance plots of other “C” values, see the appendix)

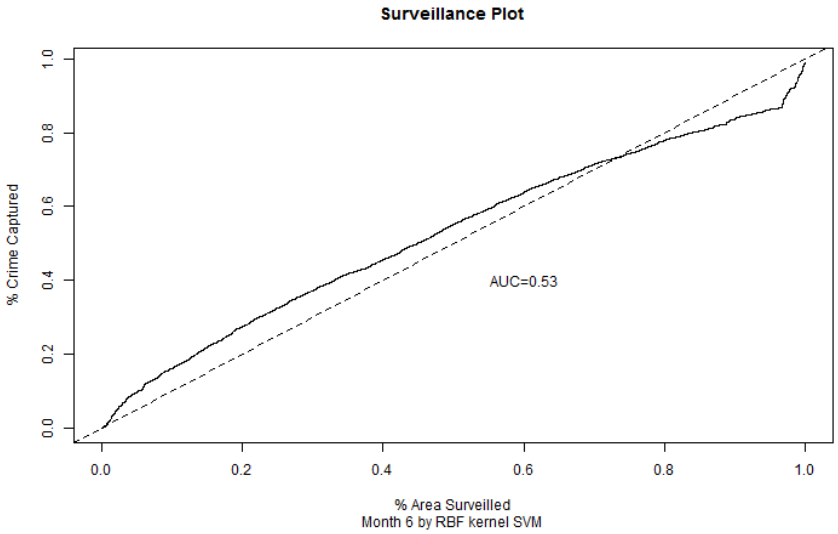


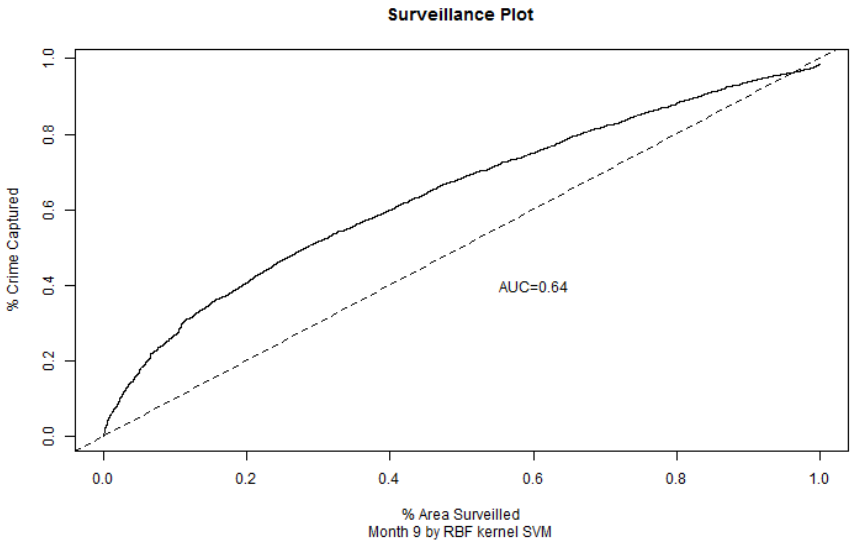




3. RBF kernel SVM (“C”=100, for surveillance plots of other “C” values, see the appendix)







4. Table for summarizing all AUC scores calculated (using different SVM models, training/testing pairs, “C” values)

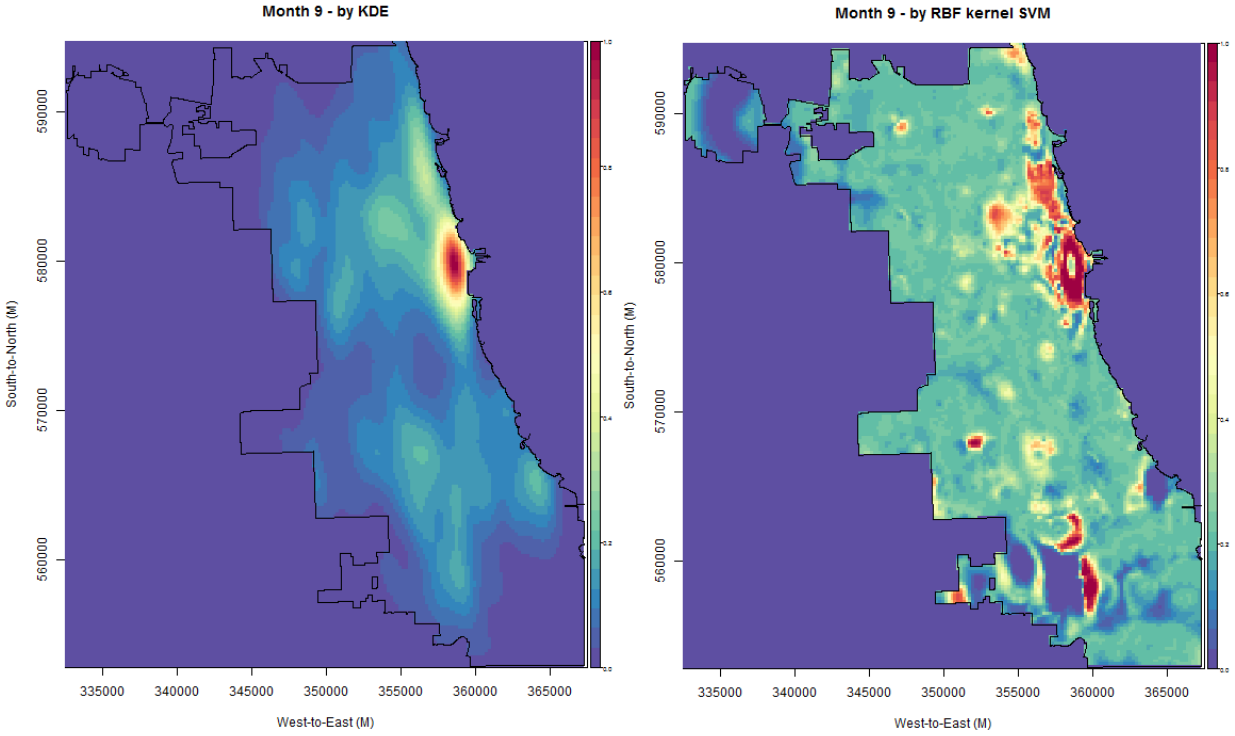
		C=100	C=10	C=1	C=0.1	C=0.01
February/March	Linear SVM	0.59	0.69	0.56	0.64	0.68
	degree-2 polynomial kernel SVM	0.64		0.62		0.62
	RBF kernel SVM	0.64		0.59		0.61
May/June	Linear SVM	0.71	0.65	0.61	0.64	0.7
	degree-2 polynomial kernel SVM	0.65		0.66		0.66
	RBF kernel SVM	0.53		0.54		0.61
August/September	Linear SVM	0.69	0.71	0.71	0.71	0.71
	degree-2 polynomial kernel SVM	0.63		0.64		0.64
	RBF kernel SVM	0.64		0.63		0.63

From the above table, we can conclude that the overall performance of the SVM models are not as good as the KDE and the logistic regression models as the majority of the AUC scores resulting from the SVM models are smaller than 0.7. The reason might be that the SVM is not a very good probabilistic model – instead it is better at classifying the data by assigning a binary response to each point. Although for each different SVM model, the dependence of AUC score on the C value seems complicated, generally speaking, for all SVM models and all training/testing pairs, a small C value (0.01) will result in a higher AUC score. This is understandable as a higher C value allows for more violations of the margin, which help fit the training data with higher accuracy, but can result in overfitting and a higher bias in the prediction. And the overall performance of the linear SVM model is better than the other two SVM models with non-linear decision boundaries (AUC scores all less than 0.66), indicating that the linear SVM model is closer to the reality. However, the overall performance of the linear SVM model (when C=0.01) is still worse than the pure KDE and logistic regression (their AUC scores are higher than 0.71). And the predicted AUC scores by linear SVM do not have a very good consistency across the different training/testing set pairs. Maybe due to the time constraints of this case study the optimized SVM model with the particular C value has not been found (so the future work may include optimizing the models by trying additional different C values and using more training/testing data sets), but the above results at least show that the non-linear SVM models may not be a good choice for describing and predicting the theft occurrence in Chicago. It is also likely that the 3 selected spatial factors (x2, x3, x4) do not actually have strong impacts on the theft crime distribution, or there are other “hidden” factors that can greatly complicate the crime patterns, which are not considered in this case study.

5. Crime density predicted by SVM models, shown in the form of heat maps

By virtue of the plot.spatial.kde() function (used in Case Study 1 and 2), and assuming the crime density value at each grid point is proportional to the probability of crime (“1”) at this location predicted by the SVM, the heat map showing the crime density could be plotted. The following is a comparison of the theft occurrence heat map of September plotted using the predictions from last month’s KDE and the RBF kernel SVM model, respectively. It can be seen that the crime

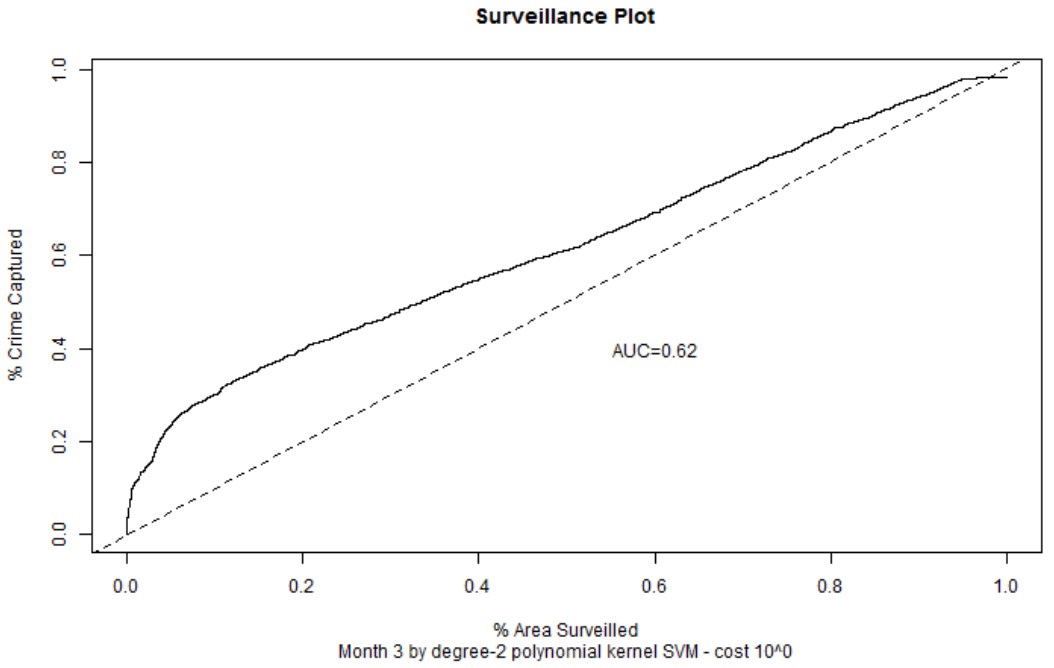
distribution predicted by SVM model looks quite different from the KDE prediction. The SVM assigned a higher theft risk to some different areas in the city, which are not indicated to have a high risk according to KDE. This also shows the relatively inaccuracy of the SVM models in predicting the theft rates.

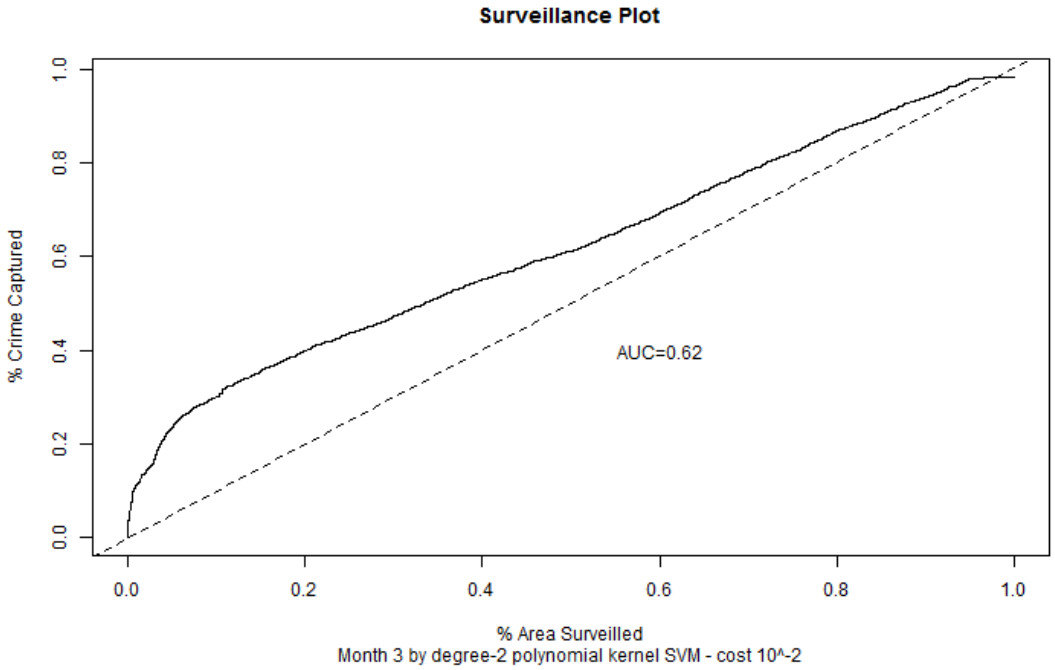


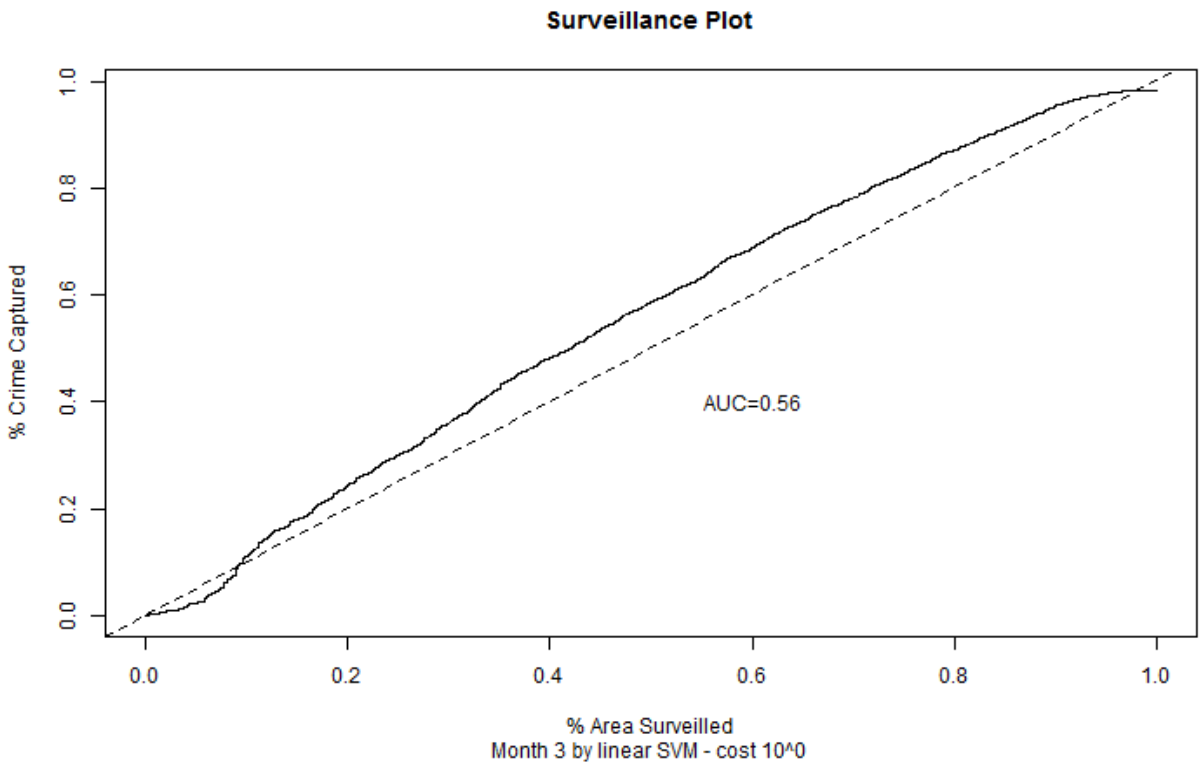
Appendix

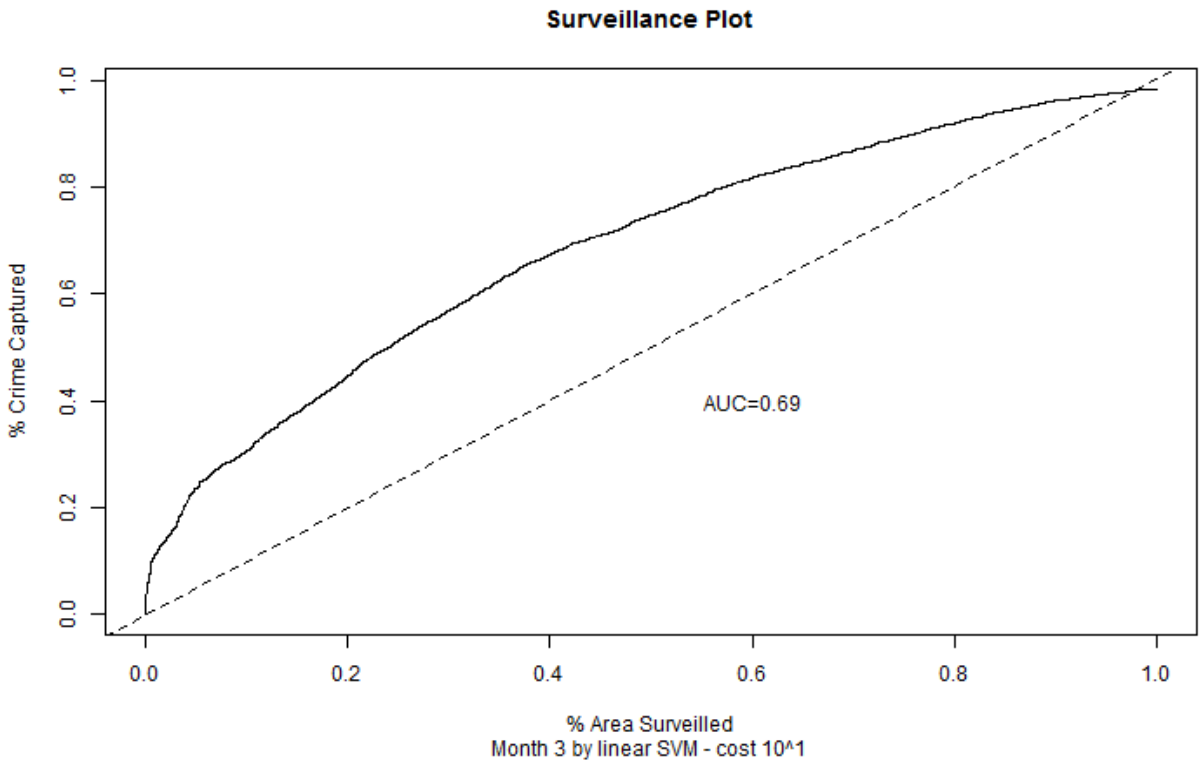
All of the rest surveillance plots generated in this case study (with “C” values different from 100):

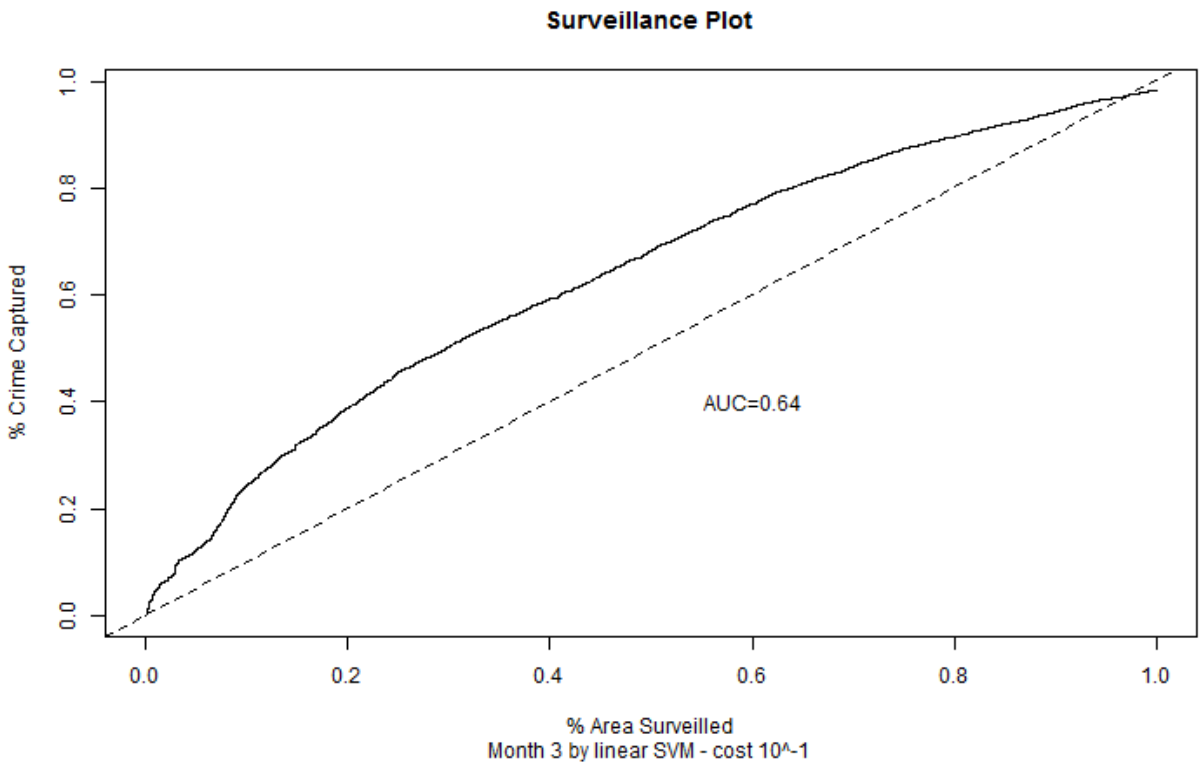
1. SVM predictions for March (using February as training set):

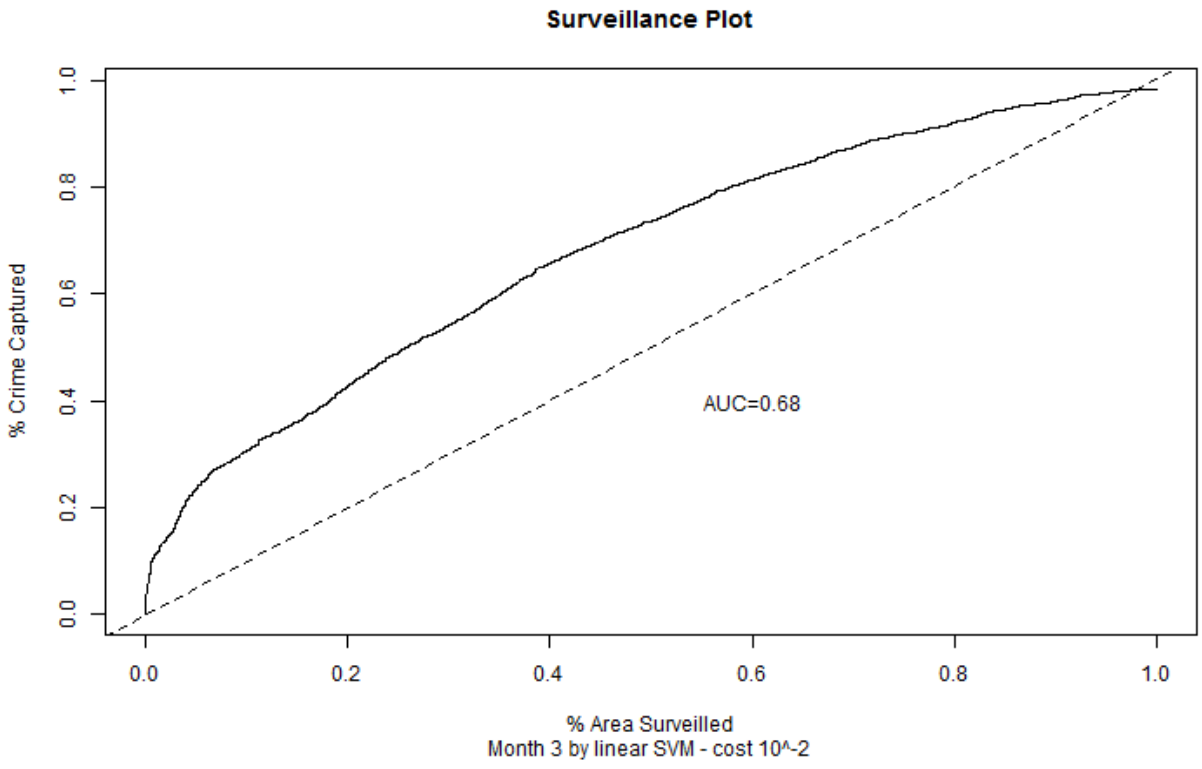


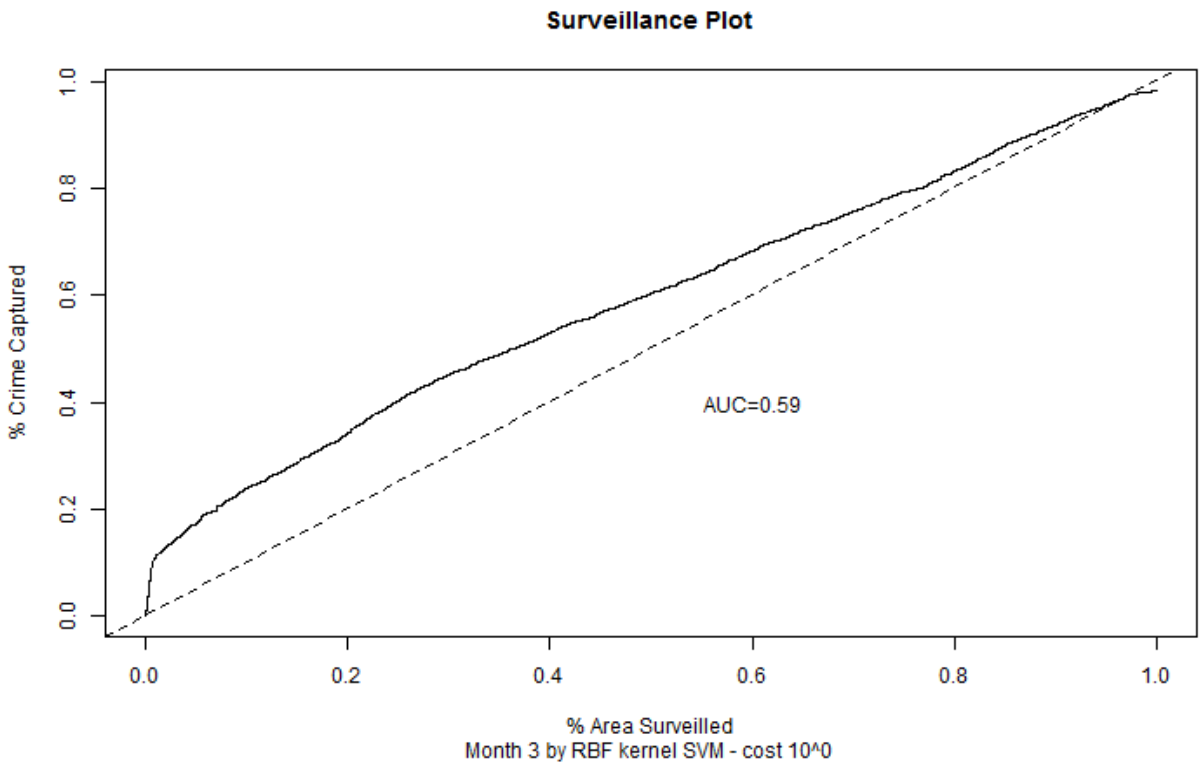


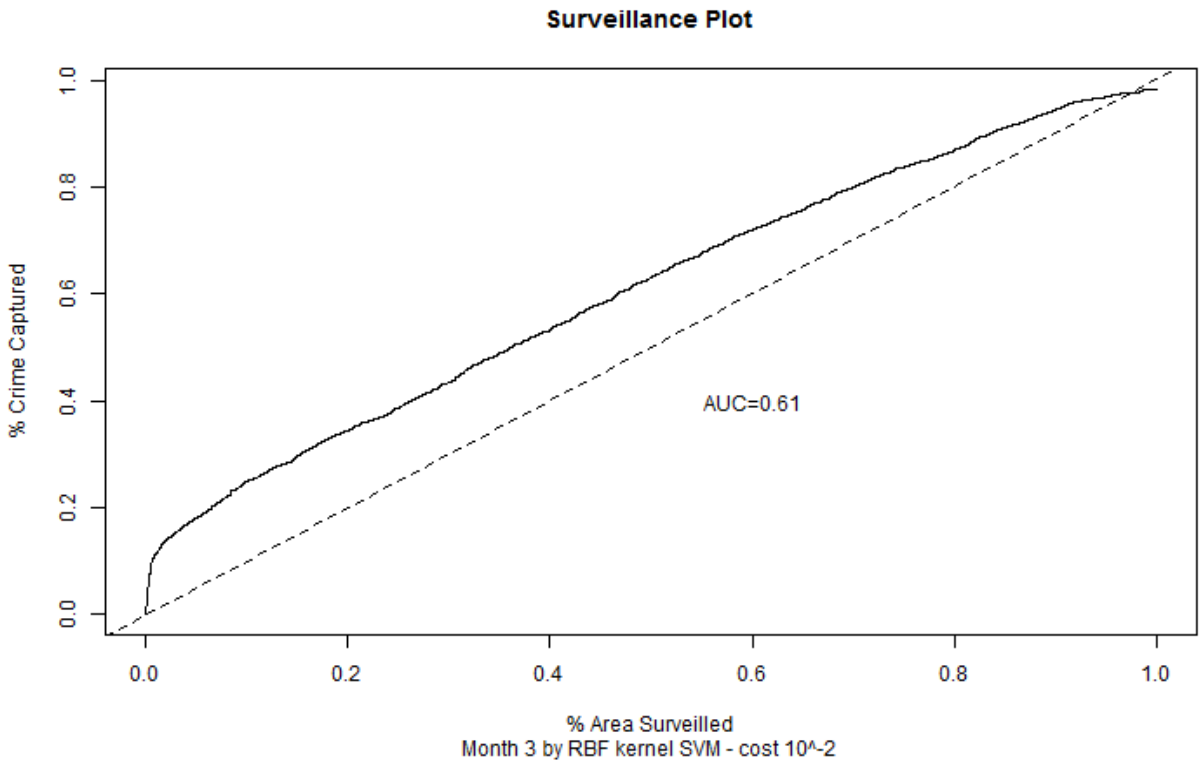

Surveillance Plot
% Crime Captured
0.0
0.2
0.4
0.6
0.8
1.0
AUC=0.56
% Area Surveilled
Month 3 by linear SVM - cost 10^0


Surveillance Plot
% Crime Captured
0.0
0.2
0.4
0.6
0.8
1.0
AUC=0.69
% Area Surveilled
Month 3 by linear SVM - cost 10^1

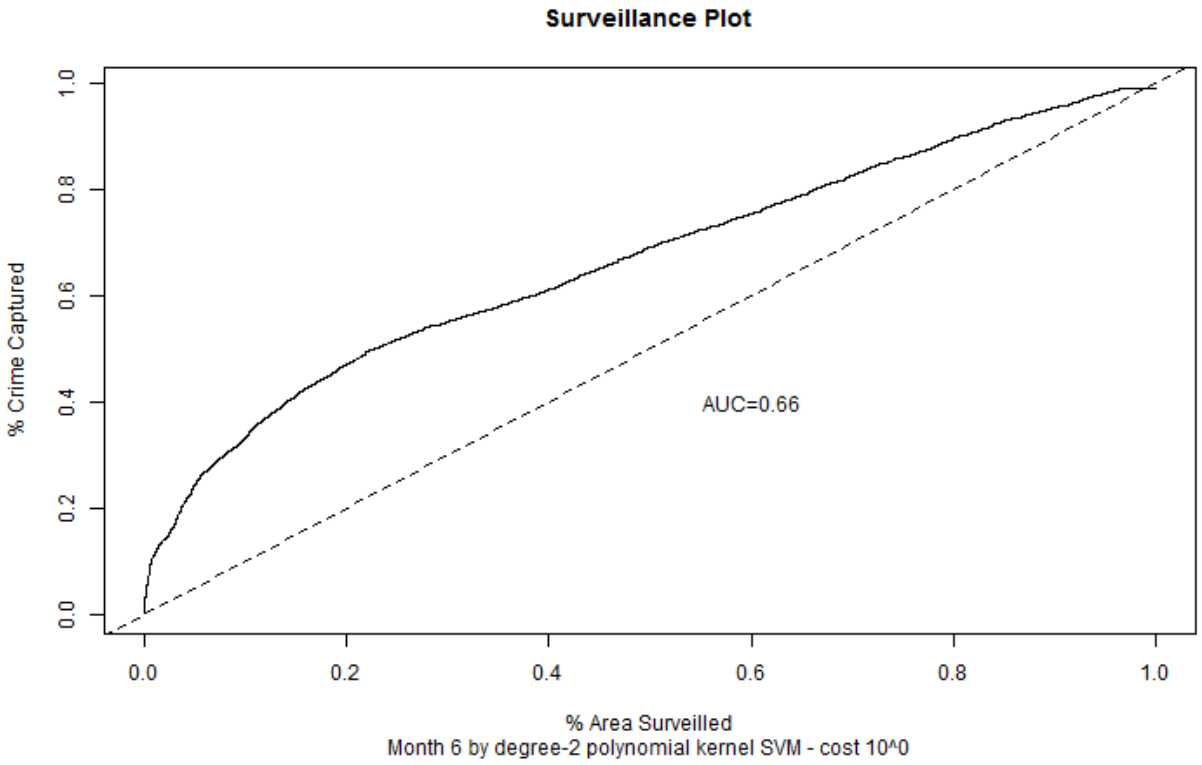

Surveillance Plot
% Crime Captured
0.0
0.2
0.4
0.6
0.8
1.0
AUC=0.64
% Area Surveilled
Month 3 by linear SVM - cost 10^-1

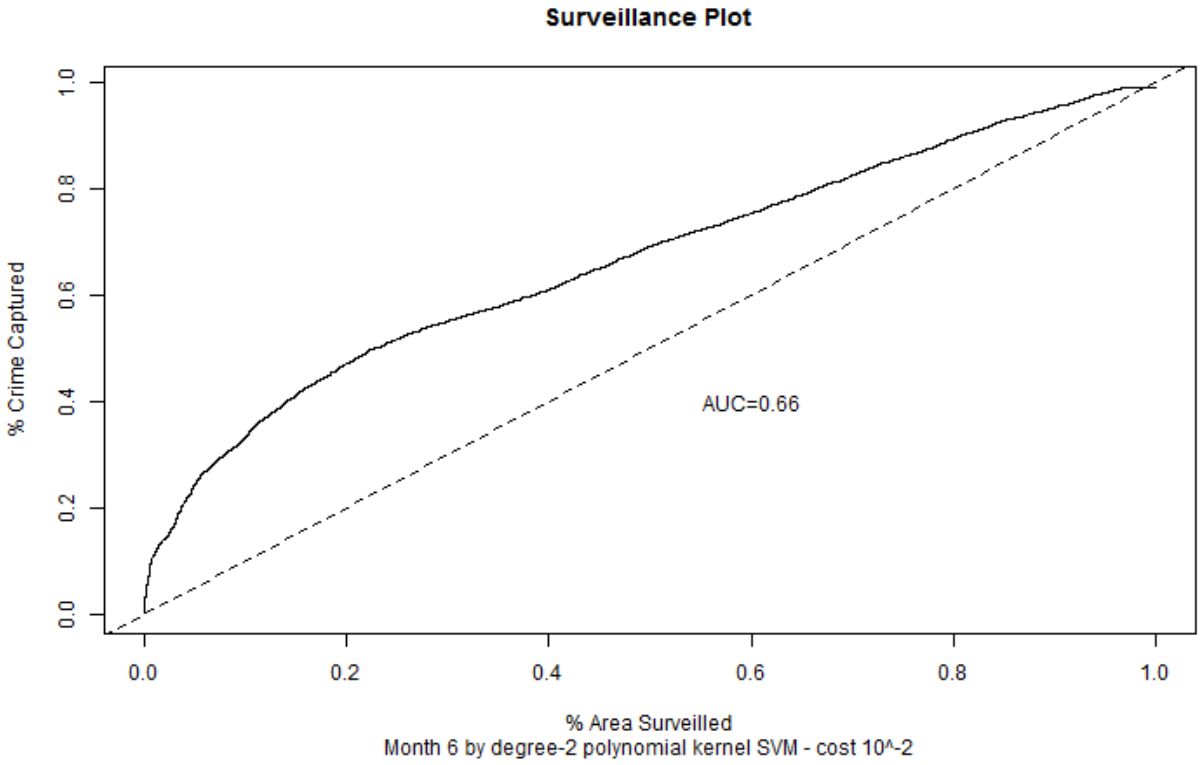

Surveillance Plot
% Crime Captured
0.0
0.2
0.4
0.6
0.8
1.0
AUC=0.68
% Area Surveilled
Month 3 by linear SVM - cost 10^-2

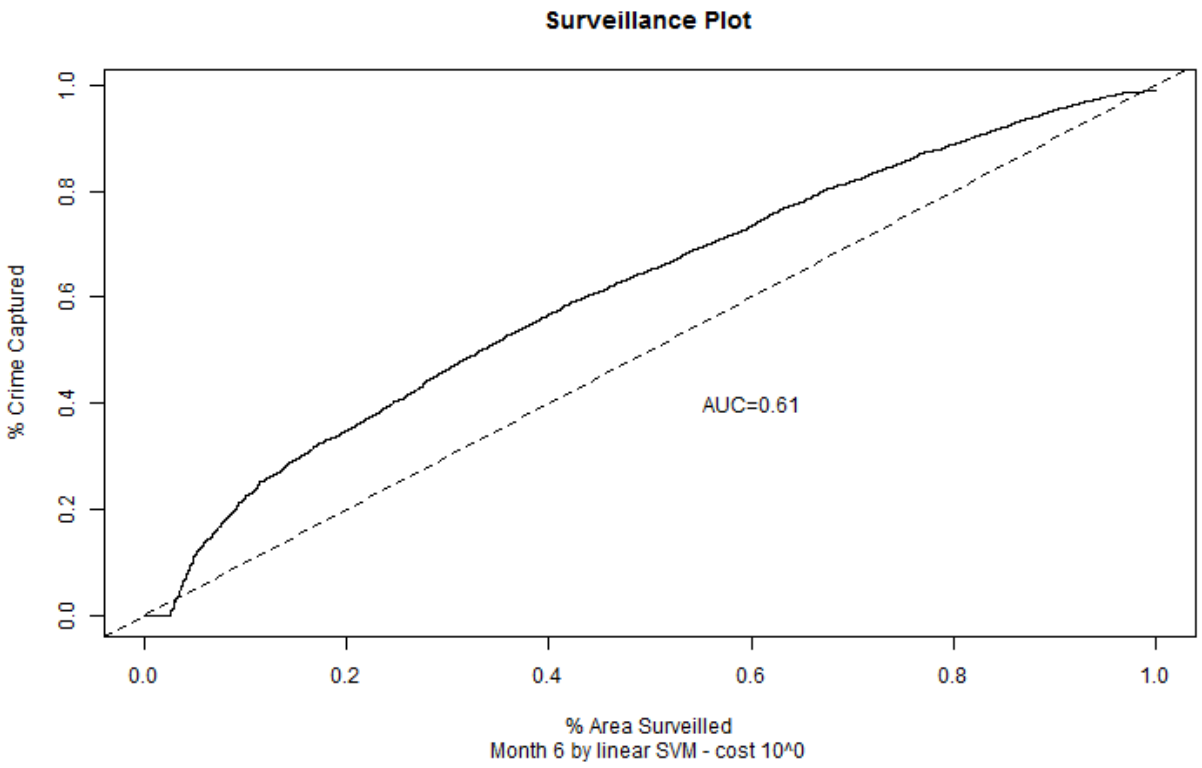

Surveillance Plot
% Crime Captured
0.0
0.2
0.4
0.6
0.8
1.0
AUC=0.59
% Area Surveilled
Month 3 by RBF kernel SVM - cost 10^0

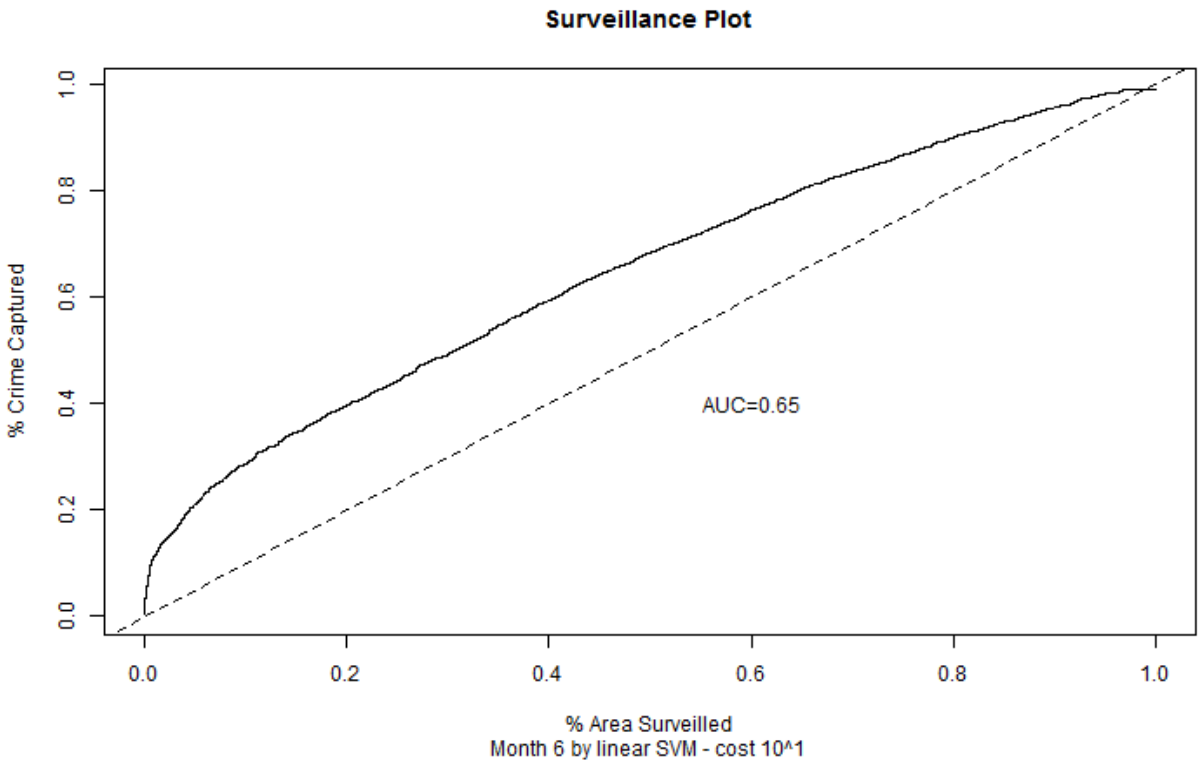

Surveillance Plot
% Crime Captured
0.0
0.2
0.4
0.6
0.8
1.0
AUC=0.61
% Area Surveilled
Month 3 by RBF kernel SVM - cost 10^-2

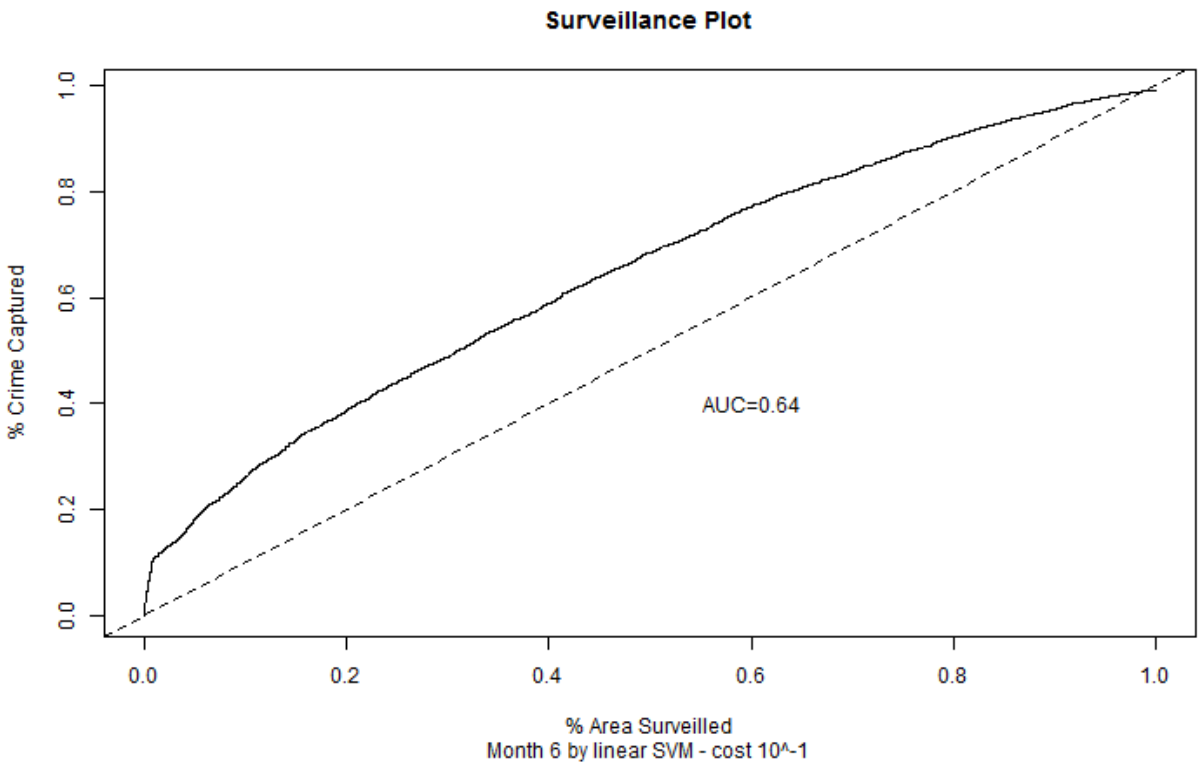
2. SVM predictions for June (using May as the training set):

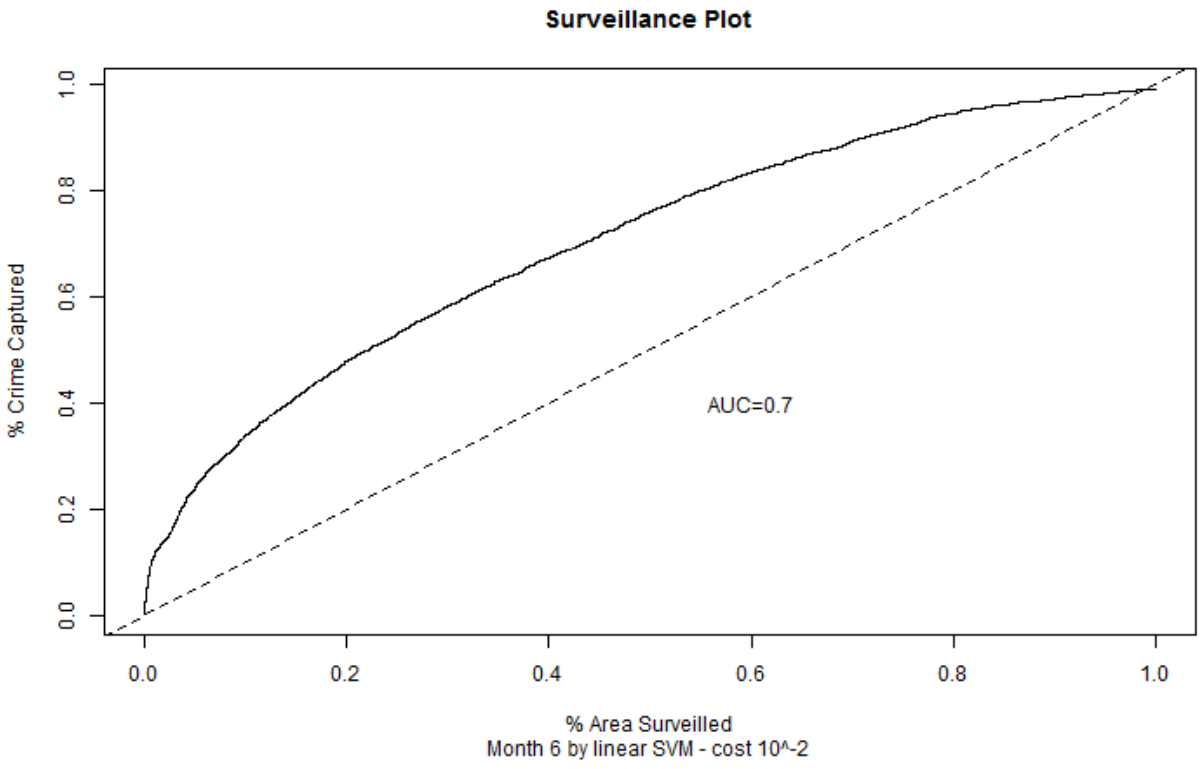


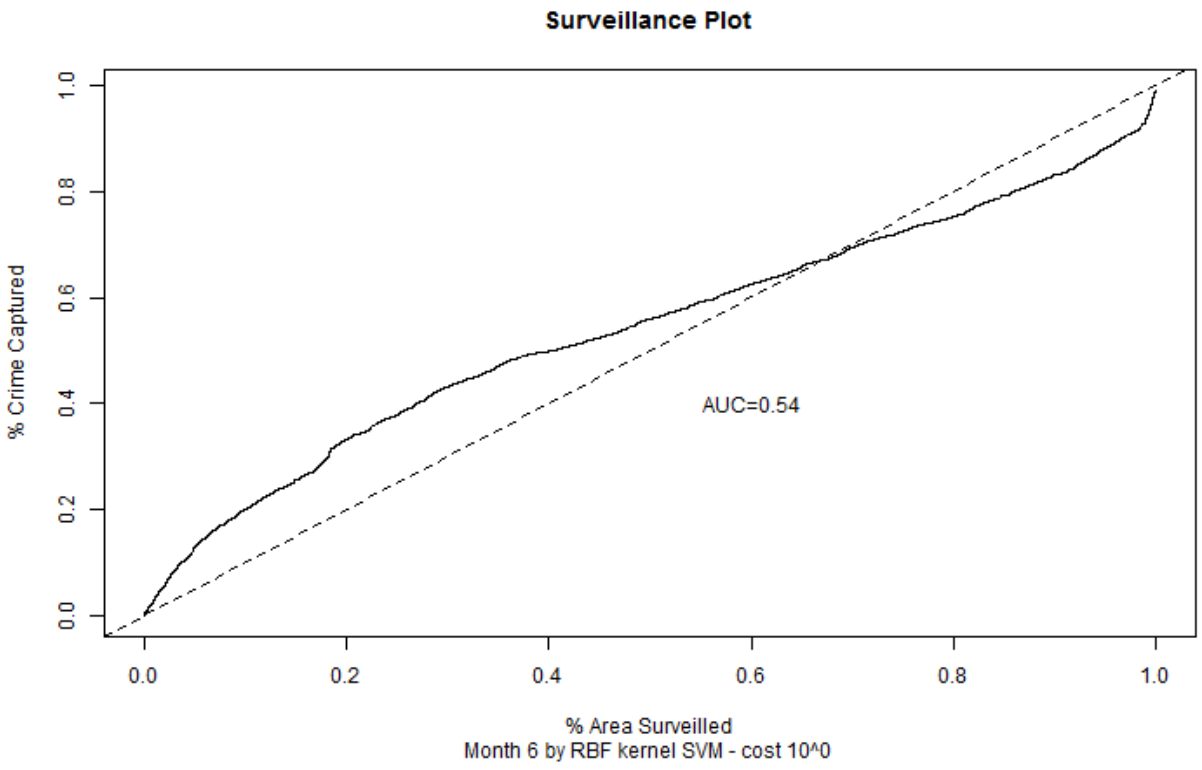


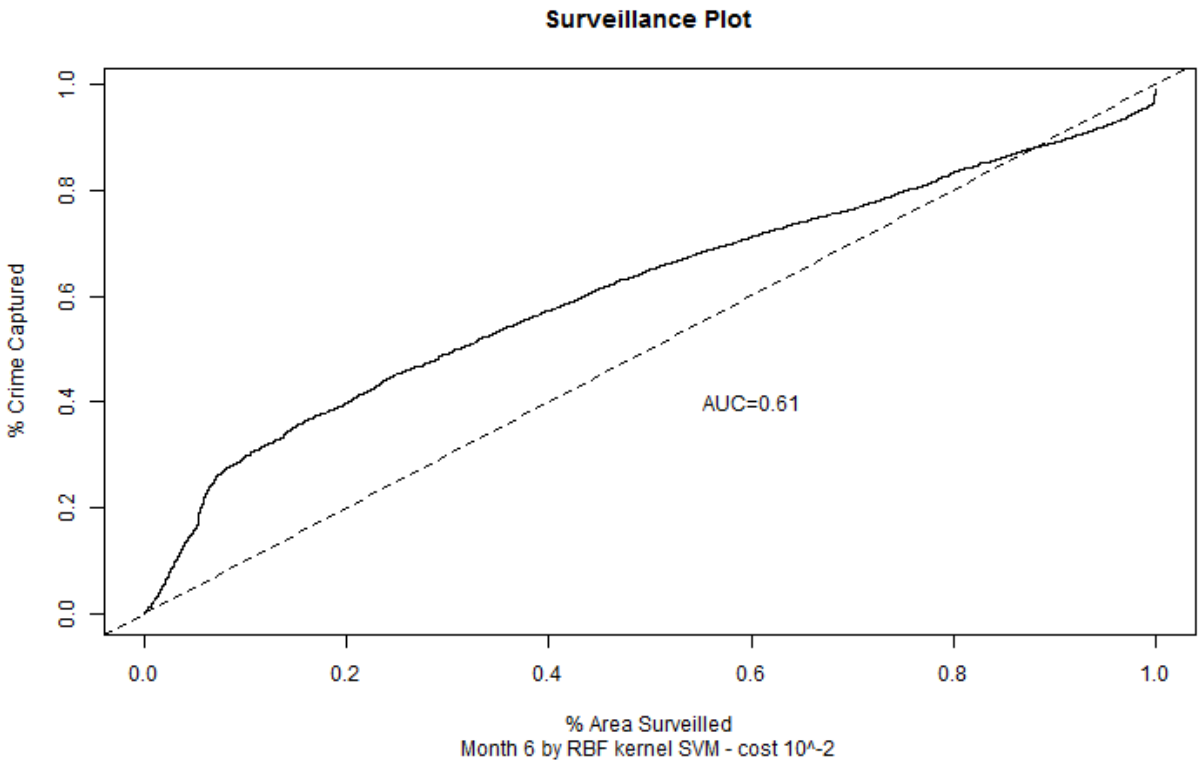

Surveillance Plot
% Crime Captured
0.0
0.2
0.4
0.6
0.8
1.0
AUC=0.61
% Area Surveilled
Month 6 by linear SVM - cost 10^0


Surveillance Plot
% Crime Captured
0.0
0.2
0.4
0.6
0.8
1.0
AUC=0.65
% Area Surveilled
Month 6 by linear SVM - cost 10^1

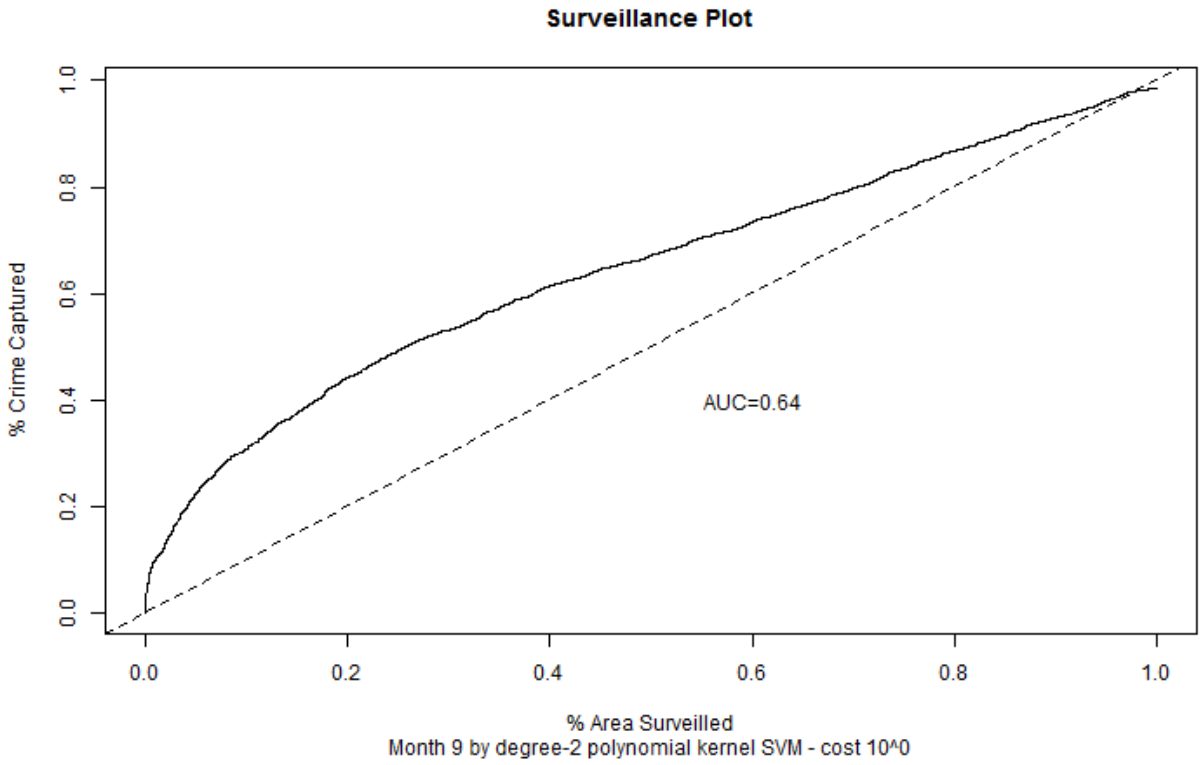

Surveillance Plot
% Crime Captured
0.0
0.2
0.4
0.6
0.8
1.0
AUC=0.64
% Area Surveilled
Month 6 by linear SVM - cost 10^-1

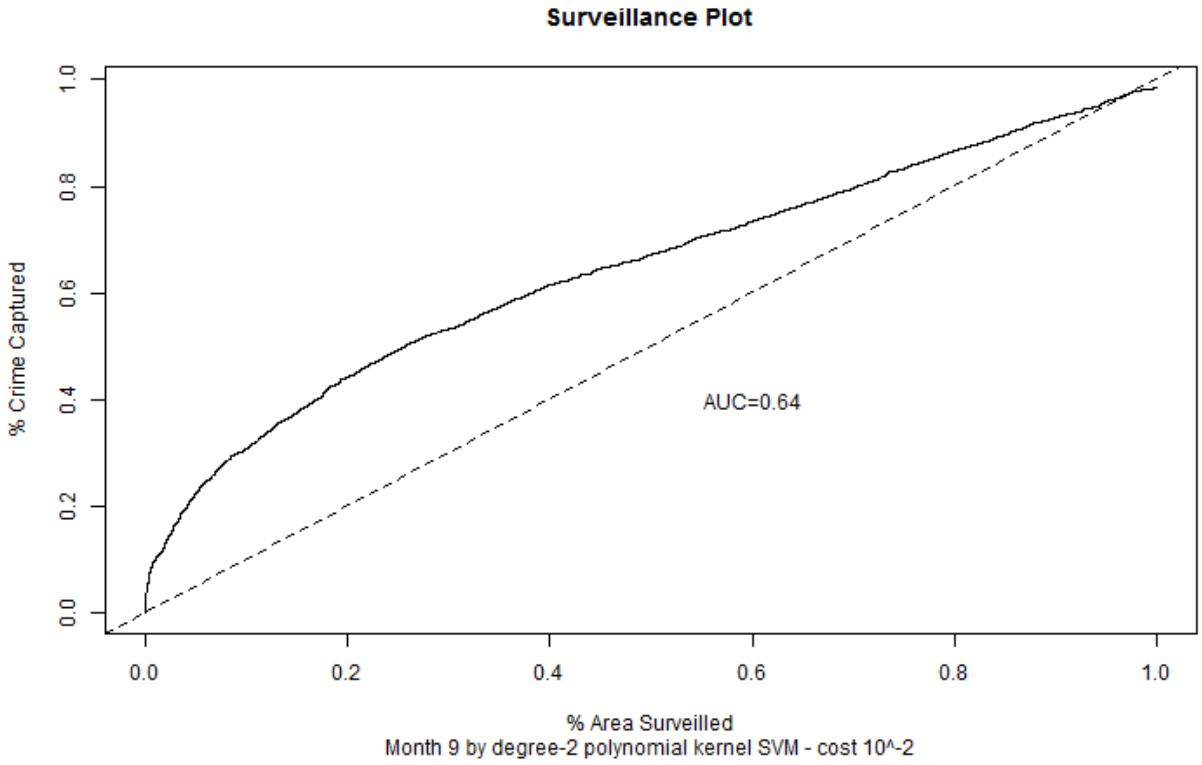

Surveillance Plot
% Crime Captured
0.0
0.2
0.4
0.6
0.8
1.0
AUC=0.7
% Area Surveilled
Month 6 by linear SVM - cost 10^-2

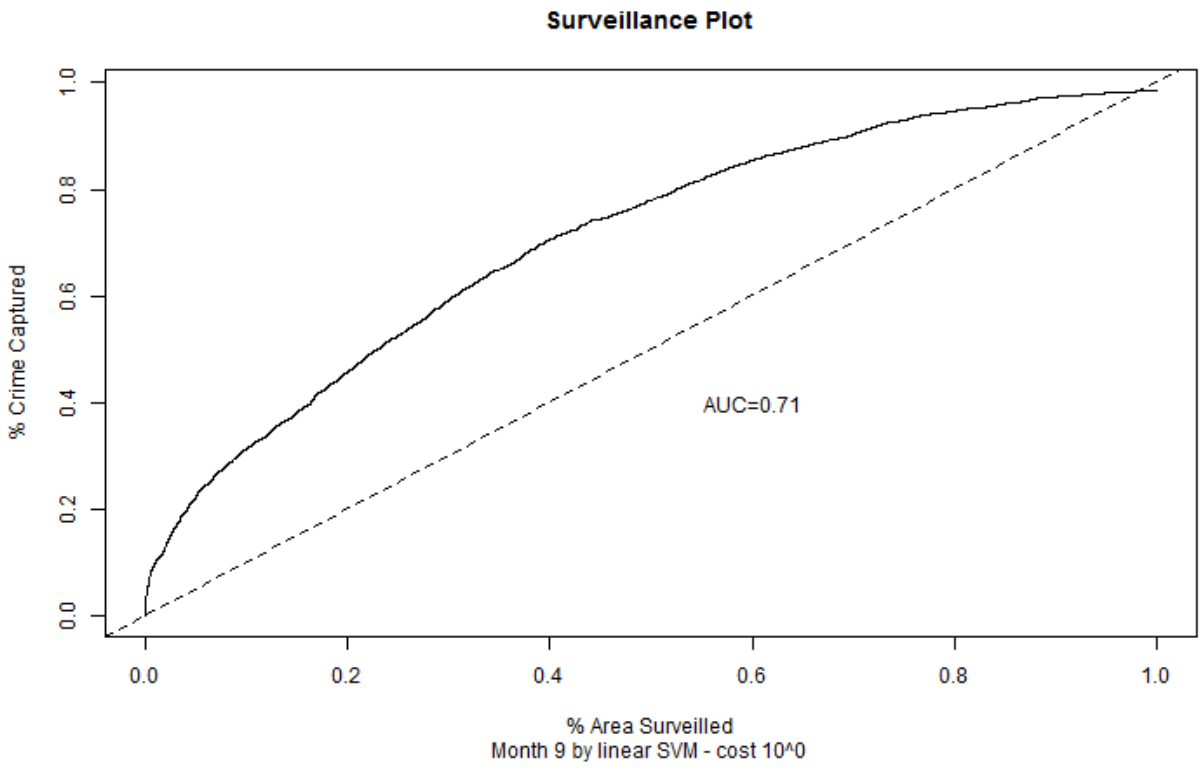

Surveillance Plot
% Crime Captured
0.0
0.2
0.4
0.6
0.8
1.0
AUC=0.54
% Area Surveilled
Month 6 by RBF kernel SVM - cost 10^0

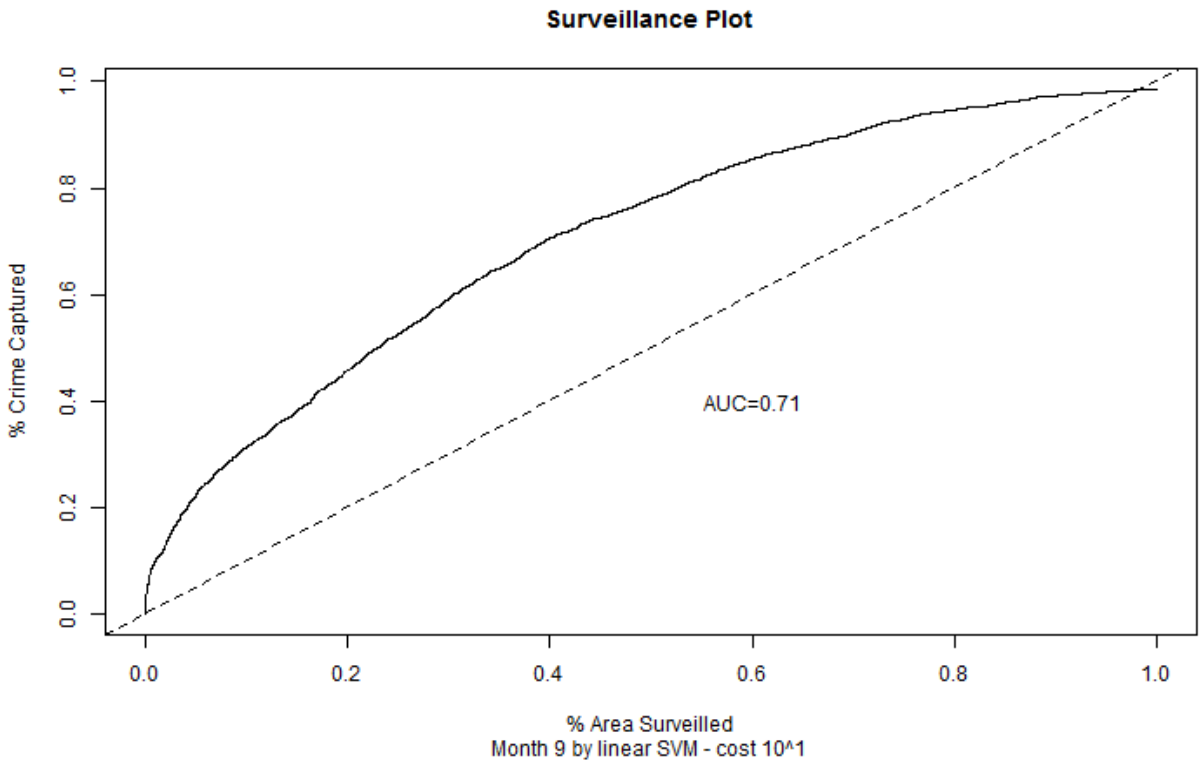

Surveillance Plot
% Crime Captured
0.0
0.2
0.4
0.6
0.8
1.0
AUC=0.61
% Area Surveilled
Month 6 by RBF kernel SVM - cost 10^-2

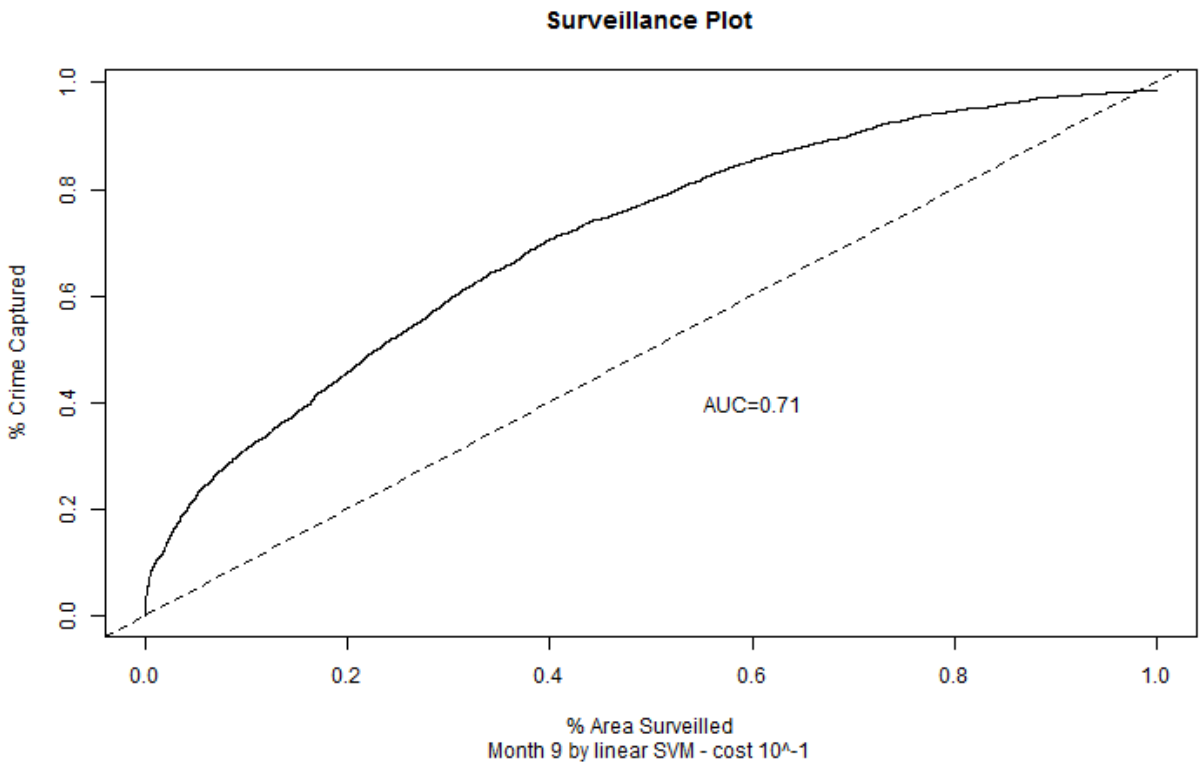
3. SVM predictions for September (using August as the training set):

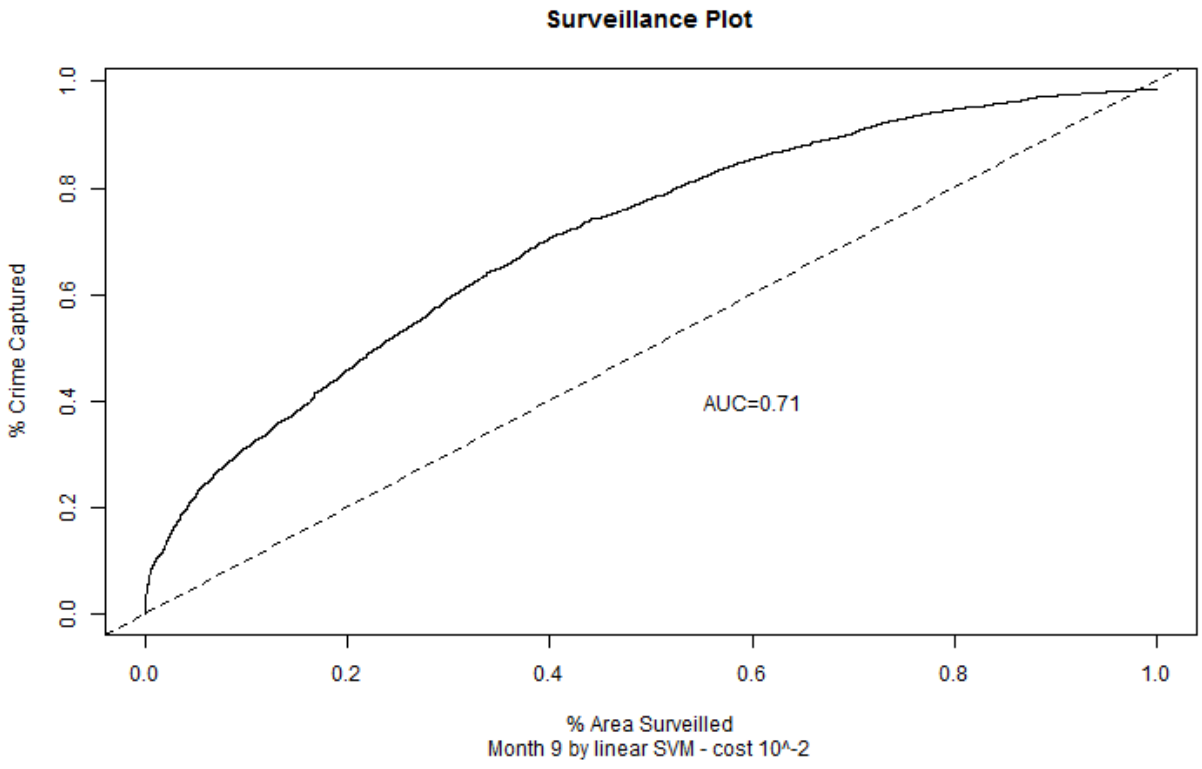


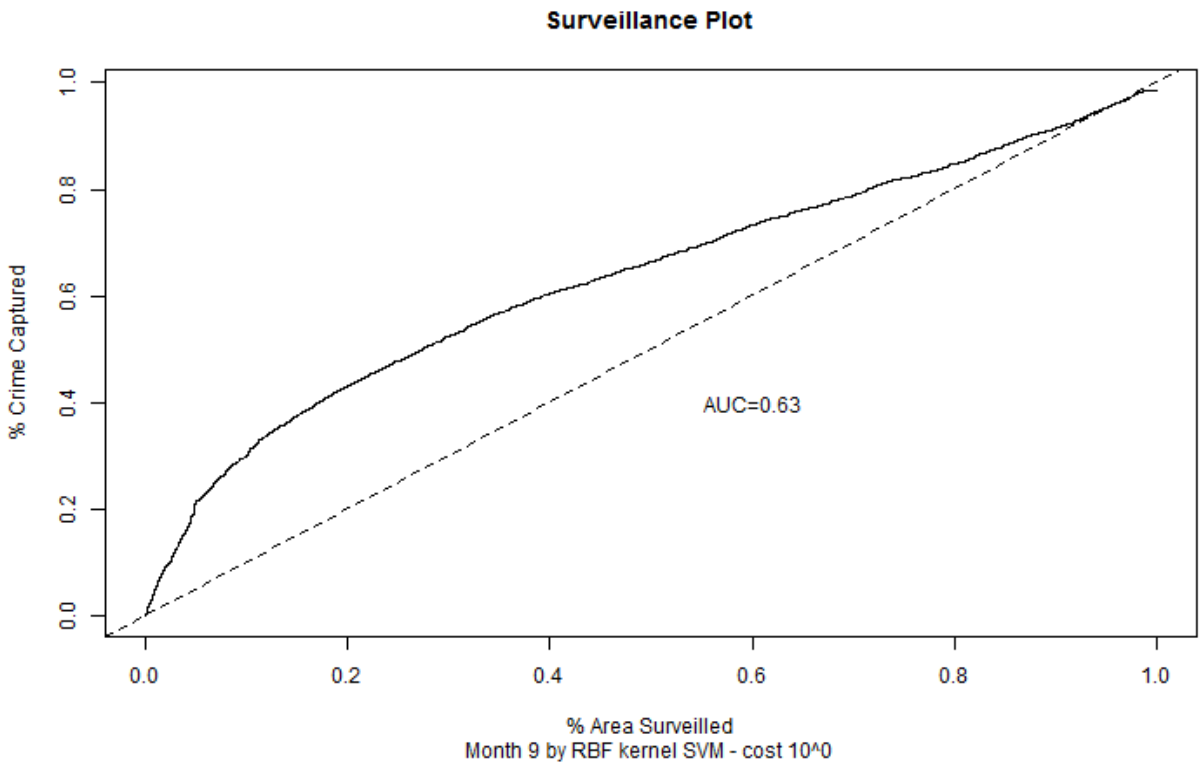


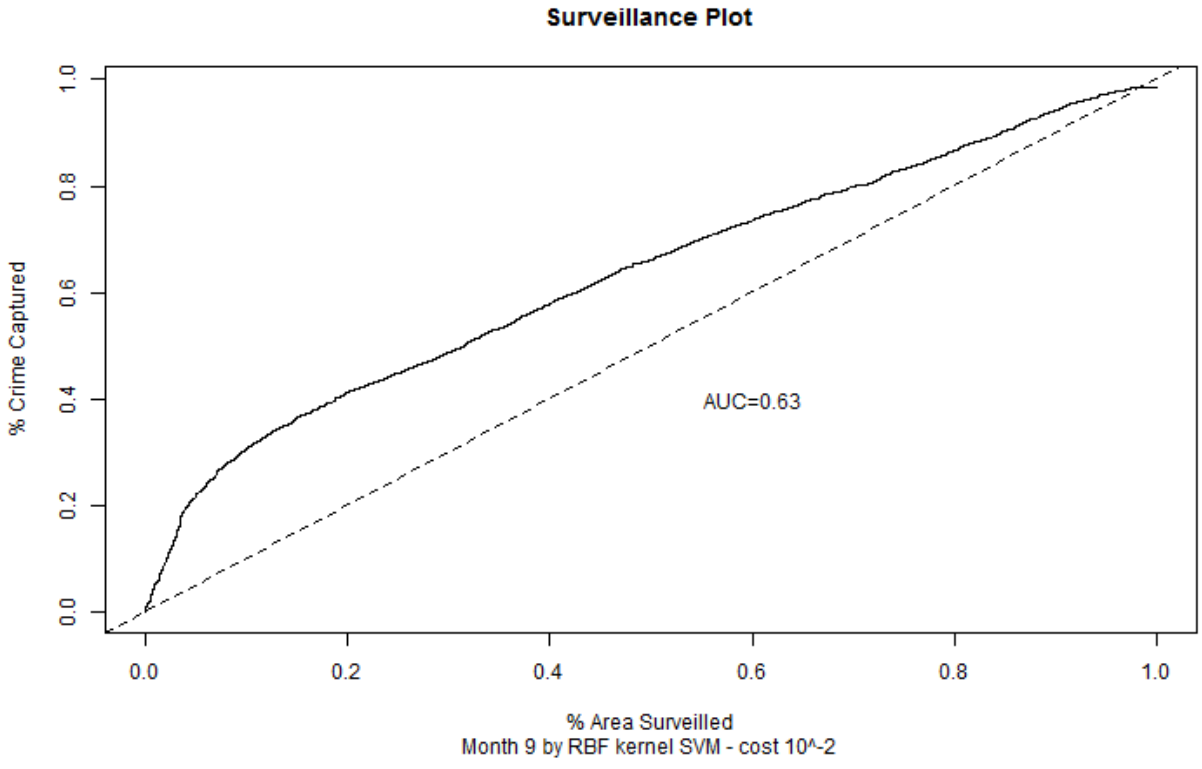

Surveillance Plot
% Crime Captured
0.0
0.2
0.4
0.6
0.8
1.0
AUC=0.71
% Area Surveilled
Month 9 by linear SVM - cost 10^0


Surveillance Plot
% Crime Captured
0.0
0.2
0.4
0.6
0.8
1.0
AUC=0.71
% Area Surveilled
Month 9 by linear SVM - cost 10^1


Surveillance Plot
% Crime Captured
0.0
0.2
0.4
0.6
0.8
1.0
AUC=0.71
% Area Surveilled
Month 9 by linear SVM - cost 10^-1


Surveillance Plot
% Crime Captured
0.0
0.2
0.4
0.6
0.8
1.0
AUC=0.71
% Area Surveilled
Month 9 by linear SVM - cost 10^-2


Surveillance Plot
% Crime Captured
0.0
0.2
0.4
0.6
0.8
1.0
AUC=0.63
% Area Surveilled
Month 9 by RBF kernel SVM - cost 10^0


Surveillance Plot
% Crime Captured
0.0
0.2
0.4
0.6
0.8
1.0
AUC=0.63
% Area Surveilled
Month 9 by RBF kernel SVM - cost 10^-2