VARIABLE SELECTION METHODS
WITH APPLICATIONS TO CRIME PREDICTION

A Thesis
Presented to the
Faculty of
California State Polytechnic University, Pomona

In Partial Fulfillment
Of the Requirements for the Degree
Master of Science
In
Mathematics

By
Joyce Mi
2016
SIGNATURE PAGE

THESIS: VARIABLE SELECTION METHODS WITH APPLICATIONS TO CRIME PREDICTION

AUTHOR: Joyce Mi

DATE SUBMITTED: Summer 2016

Department of Mathematics and Statistics

Dr. Adam King
Thesis Committee Chair
Mathematics & Statistics

Dr. Hoon Kim
Mathematics & Statistics

Dr. Jungwon C. Mun
Mathematics & Statistics
ACKNOWLEDGMENTS

I would like to express deepest gratitude to my adviser Dr. Adam King for his full support, expert guidance, understanding, encouragement throughout my study, and an excitement in regard to teaching. Without his persistent help this thesis would not have been possible. In addition, I express my appreciation to Dr. Hoon Kim and Dr. Jungwon Mun for having served on my committee. Their thoughtful questions and comments were valued greatly.

I would also like to thank my parents, sisters, brother, and my closest friends, for their unconditional love and support during the last two years; I would not have been able to complete this thesis without their continuous love and motivation.
ABSTRACT

There have been several variable selection techniques presented in the literature and coursework. However, to date there is no agreed upon “best” approach. In this thesis, we apply several different variable selection techniques to a 1990 United States crime database to determine 1) which set of socio-economic and demographic factors are most predictive of violent crime, and 2) which model selection technique produces the best out of sample predictions.

The selected cleaned data set was split into a training and test sets and stepwise (forward, backward, both forward and backward simultaneously), lasso, elastic net, principal component regression (PCR), best random subset selection (BRSS), regression trees (full, pruning, bagging, random forest) and count (aggregation of multiple methods) were fit to the training data and then applied to the test data. Fit criteria we used included AIC, BIC, multiple $R^2$, adjusted $R^2$, and mean squared error (MSE). Summary tables and figures detailing which factors were associated with violent crime were constructed as well as tables/figures assessing the performance of the different variable selection techniques.

Variables which were associated with violent crime rates across different methods of selection included measures of employment, urbanization/density, income, poverty, ethnicity and family structure. In terms of model selection techniques, the method that produced the highest out of sample $R^2$ was BRSS which also produced an unbiased assessment of model fit when applied to test data. Lasso and elastic net performed similarly, but not quite as well as BRSS. The other methods performed somewhat worse.

Predicting crime rates can be difficult, but by using these variable selection techniques we found a small subset of variables which seemed to be most associated with
crime. Most of these variables have been appeared in prior literature. Variable selection techniques vary in terms of complexity and computation time but the BRSS method seems to perform better on average. We recommend trying several model selection techniques and seeing which variables are consistently appearing as well as partitioning your data set to cross-validate your model results.
# Contents

## 1 Goal and Problem

1.1 Crime Prediction ............................................. 2

1.2 The Best Model ............................................ 7

1.3 Exploratory Data Analysis ................................. 8

## 2 Model Selection Review

2.1 Criteria for Assessing Model Performance ............... 16

2.2 Best Subset Selection ....................................... 20

2.3 Stepwise Selection ......................................... 21

2.4 Penalized Regression ....................................... 24

2.5 Dimensionality Reduction ................................. 26

2.6 Principal Components ..................................... 27

2.7 Decision Trees ............................................. 28

2.8 Bagging, Random Forests, Boosting ...................... 29

2.9 Discussion of Methods ..................................... 31

## 3 Model Selection Results

3.1 Best Random Subset Selection ............................. 34

3.2 Count and Others .......................................... 40
3.3 Regression Tree ........................................ 45
3.4 Principal Components .................................. 47
3.5 Model Performance .................................... 51
3.6 Simulation Study of 1000 Training and Test Splits ......... 54

4 Conclusion ............................................. 63

Bibliography ............................................... 66
Chapter 1

Goal and Problem

Crime rates in the United States have changed a lot since World War II. After the war, economic conditions were good and crime rates were low. However, a decade later crime rates quickly rose (early 1960s) and peaked in the 1970s-1990s. Since then, they have declined [18]. Current crime rates are similar to those seen in the 1960s [15]. Several hypotheses for the recent drop in crime include increased police presence, new legislation which gave federal funds to help fight crime (Violent Crime Control and Law Enforcement Act of 1994), expanding the prison system, decreased usage of hard drugs like crack-cocaine, legalizing abortion, and changing demographics [6]. However, crime is still a significant problem and burden on American society. In 2007, an estimate of the economic cost of crime in the United States was $15 billion in economic losses to victims and $179 billion in government expenditures [12]. Better understanding of what the potential root causes of crime are could allow us to more accurately address this problem and live in a happier and more prosperous society.

It is well known that some cities have higher crime rates than others. The causes are generally thought to be a mix of socio-demographic factors such as age, employment
characteristics, poverty rate, racial disparity, frequency of single parent households, etc. The actual cause of crime is likely multifactorial. This thesis attempts to quantify and ranks these effects through several rigorous statistical techniques. The link between socioeconomic indicators and crime rate has been well studied in the current literature. Our data set was downloaded from the UC Irvine Machine Learning Repository [16]. It includes information from the 1990 census as well as data from the 1990 US Law Enforcement Management and Administrative Statistics Survey (LEMAS) and crime data from the 1995 FBI Uniform Crime Report (UCR). Each observation in the data set is a city and for each city, various crime variables and a number of socio-demographic and other predictors variables have been recorded. To our knowledge, no paper in the current literature has attempted to utilize the data set we are using nor do the current papers apply as many rigorous statistical techniques as we are doing in this study.

We will explore several different statistical methods for predicting crime and then compare and contrast our results with 5 different countries from other papers in the literature including Canada, South Africa, Japan, Germany, and India to see if our results from the United States can be somewhat generalizable to the international community as well.

1.1 Crime Prediction

Before delving into the United States crime rate, we will first explore several other countries throughout the world to see what seems to be associated with crime rates abroad. We will investigate a country with a very high crime rate (South Africa) to one with a very low crime rate (Japan) as well as some in the middle (Canada, India, Germany). This could potentially generate further hypotheses to explore, as well as see which countries
are most similar to the US in terms of socio-demographic links to crime. If we happen
to find common general associations with crime among all countries then we can more
safely generalize our results worldwide.

Countries which have high crime rates suffer several detrimental effects. First, the
standard of living and quality of life is certainly decreased due to high crime rates[2].
Second, the countries find it hard to break the crime cycle with the next generation of
youth because criminal activity tends to hurt employment opportunities in the region
and hurt entrepreneurial activities [2]. Police in these countries tend to be generally
underfunded and overworked, which exacerbates the problem because the citizens may
think it is easier to get away with crimes.

South Africa is considered one of the most crime-stricken countries in the world
[8]. The high crime rate is generally thought to be from the unique socioeconomic sit­
uation including wealth inequality prevalent in the country. A report published by the
government in 2009 used the analyses of correlation coefficients between crime and
indicators. They found significant correlations between crime and poverty (0.44), un­
employment (0.60), and urbanization rate (0.42). They also looked into the effect of
migration/immigrants and concluded that they may contribute to crime as well but that
further investigation is needed [8].

A study from the Canadian department of justice came out in 2006 and investigated
the link between socioeconomic data (from census) and crime rates (from police depart­
ments) in two major cities. They investigated a high crime city (Saskatoon) and low
crime city (Ottawa). The statistical techniques they used were stepwise multiple regres­
sion models and principal components analysis as a variable reduction technique. They
found that the most important predictors of crime were the areas with a high percentage
of single parent households, areas with a high percentage of youth not attending school,
and areas with lower average household income. \( R^2 \) values from the stepwise regression models for the lower crime city (Ottawa) ranged from 0.269 to 0.638 for the different measures of crime, whereas in the higher crime city (Saskatoon) the same models yielded \( R^2 \) values from 0.618 to 0.726. To help mitigate this problem the paper proposes some remedies: improve social services, education, and housing opportunities for citizens [11].

India presents several unique characteristics for looking at crime rates because of large income disparity, large scale poverty, and overall a large population. Many Indian states are as large as most countries; for example, of the 15 main states, the median population was about 45 million in 1991. India is also one of the fastest emerging markets which presents an interesting case study to investigate. In a paper published in 2006 [2], Bharadwaj investigated the association between socioeconomic status and crime, and found that there was a low correlation between crime and poverty rate (0.13), unemployment (0.34), and urbanization (0.40). The \( R^2 \) from multivariate models ranged from 0.50 to 0.88 depending on type of crime and predictors used. Generally, poverty, inequality, economic growth rate of area, unemployment, population density, and education came up to be statistically significant predictors. The author concluded unemployment alone is not the driving force of crime because low paying jobs are also to blame. From a policy perspective, the author argues that people need to be brought above the poverty line with higher wages by strengthening the education system and holding education institutions accountable for upgrading the labor market, productivity, and expected incomes for individuals [2].

In a paper in 2000 [7], Entorf and Spengler looked at the association between socioeconomic and demographic factors of crime in Germany based on static regression and dynamic panel data modeling (time series analysis). They looked at proportion of
crimes cleared up by police, conviction rates, imprisonment rates, fine rates, and average length of court imprisonment. They found that ineffective policing could result in higher crime rates. They also found that East Germany tended to have higher rates, even after controlling for socioeconomic variables which the authors hypothesized were partly due to less effective policing than West Germany. In their analyses they found that foreigners and young men were most often convicted of crimes. In 1996, 28% of all suspects were foreigners, but foreigners only made up 9% of the population. In Germany, foreigners were more likely to be young men, could be fleeing their own countries for committing crimes, and maybe most come because they failed to produce economic success in their home countries. This can foster crime because the immigrants may not be as educated. They further argue that young males may be more rebellious, suffer a lack of hindsight, are in better physical position to commit crimes (more fit to run away or use intimation with speed/strength), and younger people tend to spend less time with kids/family and more with friends which can be bad influences. Men were more responsible for crimes than women, as they found that of 100 suspects, approximately 75 were male. This could be because women generally spend time with children and have fewer opportunities to commit crimes. They also found more crime in more urbanized/densely populated areas.

Japan crime rates have declined overall since WWII (homicide rate from 2.84 per 100,000 population in 1960 to 1.00 in 1990) [20]. These rates are comparatively lower than most other western industrialized countries. For example, the homicide rate in the US was 7.91 per 100,000 population in 1998. One reason for Japan’s low crime rate may be related to their cultural norms and expectations. They generally have respect for tradition, harmony, hierarchy, conformity to group norms, and seek approval from the elders. Japan is also very ethnically homogeneous and prosperous. However, the crime
that does occur appears to be disproportionately carried out by young adults and males. In a paper by Tsushima published in 1996 [20], the author found that poverty, economic inequality, and unemployment were all positively correlated with crime rates, but that it varied by type of crime. The correlation between poverty level and homicide was 0.582, but the correlation between robbery and larceny was only 0.177. Among variables used in the study, unemployment rate showed the strongest association with crime (homicide 0.687, robbery 0.516, and larceny 0.475). Their reduced models included terms for unemployment, percent young males, and a city density variable. $R^2$ for these models were 0.614 for homicide, 0.648 for robbery, and 0.508 in larceny. Note that this study used prefectures (similar to states) as unit of study because the crime rates were not large enough for analysis if cities were used, and there are 47 prefectures in Japan [20].

California is one of the largest and most diverse states in the country. The size of its economic impact is comparable to many other countries in the world. Thus, it presents an interesting case study to investigate crime in. Brown and Males [3] investigated a California crimes database from 2006 showing detailed arrest data. They found that the poverty rate displayed 3 to 4 times more explanatory power in predicting arrest rates than age. Even after accounting for ethnicity these effects still held, giving evidence to the hypothesis that poverty could be the largest contributor to higher crime rates. The dataset used for their study detailed statistics on arrests for each offense by race and Hispanic ethnicity, by age groups, ages 10-24 group, age 25-29 age group, and by 10-year age groups for ages 30 and older [3]. We will see if we find similar effects across the United States.
1.2 The Best Model

In addition to predicting crime, another main goal is to explore statistical model selection techniques such as stepwise selection (forward selection, backward selection, simultaneously forward and backward selection), lasso, elastic net, and PCA. To learn more about these techniques, refer to the chapter on Model Selection Review. We also introduce two “new” techniques count and Best Random Subset Selection (BRSS). Count is a type of model that aggregates information across the other modeling strategies. For example, after we obtain a summary of all the methods above, we will generate a list of the number of times each predictor appears in models, and count4 is a model with predictors that are selected by at least four different models. The idea for this model is that if a given predictor is being picked up by several different techniques, it is more likely to be a real signal rather than noise. Best random subset is our version of Best Subset Selection. Since best subset selection required us to fit $2^p$ models, where $p$ is the number of predictors in the predictor matrix, it is computationally intensive, especially when $p > 20$. BRSS is similar in theory, but it instead fits a subset of the $2^p$ models. If $N = 100$ is the number of randomly selected models we fit for each fixed number of $k$ of predictors $(k = 1, 2, \ldots, p)$ predictors, then we would only fit $100p$ models. As $N$ approaches $\infty$, BRSS approaches best subset selection, and if we limited $p$ to a small number, BRSS model reduces overfitting and can produce good results.

We will implement the above model selection techniques on a crime data set. The data set was first split into 75% training and 25% test sets and models were constructed using the training set and then evaluated using standard fit criteria (AIC, BIC, $R^2$, MSE) on the test set. This allows us to not only more fairly assess which technique performed “best”, but also gives us an idea for which techniques were the most biased and prone to overoptimism (e.g. high training set $R^2$ and low test set $R^2$). We will summarize which
techniques seem to be producing the most consistent and best results as well as report a standard set of factors for predicting crime which come up consistently regardless of variable selection method.

There are two main goals of model selection: prediction and making inference. Complex models tend to have adjusted $R^2$ which fit to the training data well but may not generalize well to a test data. Complex models also tend to be harder to draw causal conclusions from because of unstable parameter estimates due to multi-collinearity or lack of power. In terms of making accurate predictions, we are less concerned about the number of variables selected in the model because predictions are more robust to these issues. However, when we want to make causal claims about specific variables, we must be very careful about the selection process because coefficient estimates can vary widely and even switch signs in the presence of multi-collinearity. Whether our goal is making good predictions or making inference, the ideal model selection method would result in a model that has accurate predictions and coefficients that are easier to interpret for causal conclusions. In other words, we want a simple model that not only makes accurate prediction but also easy for making inferences.

### 1.3 Exploratory Data Analysis

The data set is from the 1990 United States crime database from the UC Irvine Machine Learning Repository. The original data set contains 147 variables and 2,215 observations (cities), which includes unrelated attributes such as county code. After excluding these unrelated attributes, cities with missing variables, other predicted variables, and variables that are linear combinations of other variables, the cleaned data set had 95 variables and 807 observations. This is the data set we will be use for this study. The predicted
variable \( Y \), violent crime is measured per capita (per 100,000 population) for the number of violent crimes in a community. The predictor matrix \( X \) included income, family, ethnicity, number of children, education, percentage of household with two parents, etc.

The first thing we did to determine potential important predictors of violent crime was to construct a full model (model with all 94 predictors in the Complete Data set) and assess pairwise correlations between crime and all the predictors. We report the variables that met the conditions of \( p \) value \( \leq 0.05 \) for estimates of the full model and \( |r| > 0.3 \) for the correlation coefficient in Table 1.1. We found thirty three predictors that were associated with crime given this relatively arbitrary criteria. Among other predictors, the percentage of population that is African American is positively correlated with violent crime where the percentage of population that is White, is negatively correlated with violent crime. However, this is not evidence to make a causal conclusion because there are other confounding variables that are not accounted for in this simple analysis. Some of

<table>
<thead>
<tr>
<th>Full Model</th>
<th>Correlation Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predictor</td>
<td>Estimate</td>
</tr>
<tr>
<td>racepctblack</td>
<td>18.47</td>
</tr>
<tr>
<td>agePct12t29</td>
<td>-58.09</td>
</tr>
<tr>
<td>pctWage</td>
<td>-23.29</td>
</tr>
<tr>
<td>PctPopUnderPov</td>
<td>-19.07</td>
</tr>
<tr>
<td>PctKidsBornNeverMar</td>
<td>42.33</td>
</tr>
<tr>
<td>PersPerOccHous</td>
<td>1697</td>
</tr>
<tr>
<td>PersPerRentOccHous</td>
<td>-531.7</td>
</tr>
<tr>
<td>HousVacant</td>
<td>0.01758</td>
</tr>
<tr>
<td>RentLowQ</td>
<td>-1.36</td>
</tr>
<tr>
<td>MedOwnCostPctIncNoMtg</td>
<td>-47.28</td>
</tr>
<tr>
<td>adj. R2</td>
<td>0.6862</td>
</tr>
<tr>
<td>mul. R2</td>
<td>0.7228</td>
</tr>
<tr>
<td>p-value</td>
<td>&lt;2.2e-16</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
the predictors that have high correlation coefficients with violent crime are: percentage of kids born to never married parents, percentage of households with public assistance income, percent of vacant housing that is boarded up, percentage of population who are divorced, percentage of people under the poverty level, and percentage of females who are divorced. Predictors that are negatively correlated with violent crime are: percentage of kids with two parents, percentage of families with two parents, percentage of teens with two parents, percentage of young kids with two parents, and percentage of households with investment or rent income. In other words, communities with more families that have two parents, wealthy, or white race are associated with lower crime rates whereas communities with more single parents, higher percentages of families with public assistance, and black race are associated with high violent crime rate.

Based on the full model and the correlation table, there are over thirty predictors that are highly associated with violent crime. Table 1.2 are univariate summary statistics of some of the selected variables and Figures 1.2, 1.3, and 1.4 are boxplots of these variables. There are many predictors with extreme outliers, and some of these variables are: number of kids born never married parents, number of people under the poverty level and number of vacant households. Note that these outliers are meaningful since there are communities with large number of people under the poverty line whereas some other well off communities might have a much smaller number of people under the poverty level. Note that there also communities with much higher violent crime rates per 100,000 of people. Some of the cities with high crime rates are Chester (PA), Alexandria (VA), Camden (NJ), Atlantic City (NJ), Newark (NJ), Anniston (AL), Atlanta (GA) and Miami (FL), with violent crimes (per 100,000) of 4877.06, 3530.78, 3834.1, 3583.48, 3928.03, 3540.57, 4026.59 and 3829.21, respectively. Communities with much smaller violent crime rates were: Harvardtown (MA), Simsburytown (CT), Mequon city (WI), Cranber-
rytownship (PA), Warren town (RI), Ogdensburg city (NY), Polandtownship (OH) and Oswego city (NY), with respective crimes (per 100,000 population) 7.79, 9.07, 9.24,
Figure 1.1: Population and Violent Crime

- The dashed line is the mean of violent crime.
- Points above the solid line are outliers (above the third quartile).

6.64, 8.82, 7.6, 8.85, and 5.35.

Figure 1.1 is a plot of population and violent crime. The blue solid line is the outlier line (above the third quartile) and the dashed line is the mean.
Figure 1.2: Boxplots of Selected Variables

- MedOwnCost
- PctIncNoMtg
- NumKidsBorn NeverMar
- PctKidsBorn NeverMar
- PersPer RentOccHous
- FemalePctDiv
- MalePctDivorce
- MalePctNevMarr
- TotalPctDiv
- PctFam2Par
- PctKids2Par
- PctTeen2Par
- PctYoungKids2Par
- medFamInc
- medIncome
- perCapInc
- pctWInvInc
Figure 1.3: Boxplots of Selected Variables

- PctPopUnderPov
- NumUnderPov
- pctWPubAsst
- PctUnemployed
- PctHousOwnOcc
- HousVacant
- PctPersOwnOcc
- PersPerOccuHous
- PctHousLess3BR
- PctHousNoPhone
- PctPersDenseHous
- PctWOFullPlumb
- PctLess9thGrade
- PctNotHSGrad
- agePct12t29
- PctEmploy
Figure 1.4: Boxplots of Selected Variables

- **pctW Wage**
- **RentLowQ**
- **racepctblack**
- **racePctWhite**
- **PctVacantBoarded**
- **ViolentCrimesPerPop**
Chapter 2

Model Selection Review

Regression literature on model selection and prediction generally assumes that the true model is one of the form

\[ Y = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p + \varepsilon \]  

(2.1)

Our goal is to predict \( Y \), our response variable using \( X \), our predictor matrix with \( p \) variables. In this case, a key issue is to identify variables that are not related to \( Y \). The goal here is to identify nonzero \( \beta \)'s. This task is to build a model of the form (2.1) [17]. Several methods for selecting optimal models have been proposed in the statistical literature. Some of the most commonly used methods are: stepwise procedures, best subset selection, shrinkage, and dimensionality reduction [17, 9].

2.1 Criteria for Assessing Model Performance

Below is a review of the four classes of criteria used in model selection [17].

1. Goodness-of-fit Criteria
Residual sum of squares (RSS) is a measure of goodness of fit. However, it does not account for the number of variables in a model. Therefore, the estimated residual variance is a better measure of goodness of fit. This is given by

\[
S^2 = \frac{RSS}{n - p} \tag{2.2}
\]

where \(p\) is the number of \(\beta\)'s we’re estimating. Another criterion is the \textit{multiple r-squared} or \(R^2\), also known as the coefficient of determination. It measures the proportion of total variation of response variable explained by a model. It is given by

\[
R^2 = 1 - \frac{RSS}{TSS} \tag{2.3}
\]

Models with the maximum multiple \(R^2\) are chosen as the optimal model when used as a criteria for model selection. \(TSS\) is the sum of the squares of the difference of the response variable and its mean, and it is given by

\[
TSS = \sum_{i=1}^{n} (y_i - \bar{y})^2. \tag{2.4}
\]

However, multiple \(R^2\) does not account for the number of predictors \(p\) in models, and it increases as the number of predictors in a model increases. Adjusted \(R^2\) is a better alternative, and it is defined by

\[
R_a^2 = 1 - (1 - R^2) \frac{n}{n - p} \tag{2.5}
\]

Similar to multiple \(R^2\), models with the maximum adjusted \(R^2\) are chosen as the optimal model when used as a criteria for model selection. The quantity can loosely be interpreted as the estimated proportion of variation in the response explained by the model with a small penalty for including extra predictors relative to the sample size. Thus, \(R_a^2\) penalizes a model for using extra predictors.
2. Criteria Based on Prediction Error

Model error (ME) measures how well a particular model predicts external data, and it is given by

\[
E \left( \sum (\hat{Y}_j - E[Y_j|X_j]) \right)^2
\]

However, since expected ME involves the unknown mean \( \mu = E[Y|X] \), it cannot be used as a criterion for model selection. We can use the estimate of the expected value of \( \frac{ME}{\sigma^2} \), known as the Mallow's \( C_p \), as an operational criterion for model selection, where \( C_p \) is given by

\[
C_p = \frac{RSS_p}{\sigma^2} + 2p - n
\]

\( RSS_p \) is the residual sum of squares from fitting a model with \( p \) parameters and have \( n \times p \) prediction matrix \( X_p \). Since \( C_p \) estimates the scaled expected ME, choosing the model with smallest \( C_p \) may result in good predictions. Cross-validation is another way to estimate prediction error. The idea of cross-validation involves partitions observations (rows) into two disjoint subsets, perform analysis (e.g least squares) on one subset (called training set), and validate the analysis on the other subset (called test set). To reduce variability and the dependence of a given split, multiple rounds of cross-validation are performed using different partitions, and then the average of the performance across all partitions is calculated. For example, if \((X_i, Y_i)\) is the training set, and \((X_{0i}, Y_{0i})\) where \(i = 1, 2, \ldots, m\) is the test set, then the average squared prediction error is given by

\[
\frac{1}{m} \sum_{i=1}^{m} \left( Y_{0i} - \beta_0x_{0i} \right)^2
\]

where \( \beta \) is calculated using the training data set [17]. A third method for estimating the prediction error (PE) or ME of the model selected is the little bootstrap. Seber
states that this method gives an almost unbiased estimate of the PE of a selected model [17].

3. Estimating Distributional Discrepancies

Another type of criterion measures the distance between the true model of the distribution of data set \( Y \) and a candidate model. Let \( f \) denote the true model and \( g \) denote the candidate model, and \( I(f, g) \) (Kullback-Leibler, K-L information) is the information lost when we use \( g \) to approximate \( f \). The K-L information is given by

\[
I(f, g) = \int f(x) \log \left( \frac{f(x)}{g(x|\theta)} \right) dx
\]

(2.9)

The criterion \( I(f, g) \) cannot be used directly in model selection because the true model is unknown. However, we can use the estimate of the K-L information, Akaike’s information criterion (AIC), for model selection. AIC is given by

\[
AIC = -2\log L(\hat{\theta}) + 2K
\]

(2.10)

where \( K \) is the asymptotic bias correction term, and \( L(\hat{\theta}) \) is a maximum of likelihood function. Since Akaike’s procedures for model selection are based on K-L information and it has certain cross-validation properties, it’s a reasonable criteria for model selection. When \( K \) is large relative to sample size \( n \), there is a small-sample version called \( AIC_c \), and it’s given by

\[
AIC_c = -2\log \left( L(\hat{\theta}) \right) + 2K + \frac{2K(K+1)}{n-K-1}
\]

(2.11)

Note that as \( n \) gets larger, \( AIC_c \) converges to \( AIC \). Thus, \( AIC_c \) should be used unless \( \frac{K}{n} > 40 \) [4, 17]. When \( \sigma^2 \) is known, AIC is given by

\[
AIC = \frac{RSS_p}{\sigma^2} + 2p
\]

(2.12)
where \( p \) is the number of parameters. Note that if \( \sigma^2 \) is replaced by \( \hat{\sigma}^2 \), AIC is different by a constant \( n \) (see equation 2.5) compared to \( C_p \). Thus, AIC is very similar to \( C_p \). A generalization of this version of the AIC is called Bayesian information criterion (BIC) is given by

\[
BIC = \frac{RSS_p}{\sigma^2} + a_n P
\]  

(2.13)

where \( a_n = \log n \), which depends on \( n \). BIC is motivated by Bayesian theory, but it has a non-Bayesian interpretation, similar to AIC. Both are goodness-of-fit measures that penalize models that have an excessive number of parameters in them. For sample sizes larger than 7, (when \( \log n > 2 \)), the BIC imposes a greater penalty for each additional parameter than does the AIC [17].

### 2.2 Best Subset Selection

Best subsets is also known as the “all possible regression method” which requires the fitting of \( 2^p \) possible models where \( p \) is the number of predictors. One unique characteristic of this method is that it evaluates the criteria for all possible subsets of predictors. The goal here is to exclude irrelevant variables from a multiple regression model and identify the “best” subset of predictors that are related to \( Y \). This is also known as the subset selection [9, 17].

To perform best subset selection, we fit a separate least squares regression model for each possible subset of \( p \) predictors. We then look at all of the resulting models, pick the one that’s “best” based on the four classes of criteria described above. This procedure is usually broken up into two stages [9].

1. Let \( M_0 \) denote the intercept model, which contains no predictors.
2. For $k = 1, 2, \ldots, p$,

(a) Fit all $\binom{n}{k}$ models that contain exactly $k$ predictors.

(b) Pick the best among these $\binom{n}{k}$ models, and call it $M_k$. Here the best is defined as having the smallest RSS, or largest $R^2$.

3. Select a single best model among $p + 1$ models: $M_0, M_1, \ldots, M_p$, using cross-validated prediction error, $C_p$, $AIC$, $AIC_c$, $BIC$, or adjusted $R^2$ [9, 17].

### 2.3 Stepwise Selection

Best subset selection can be computationally intense and not possible for a large data set. This method may also suffer from statistical problems such as overfitting and high variance of the coefficient estimates when $p$ is large. For both of these reasons, the stepwise methods, which explore a restricted set of models, may be good alternatives [9]. They are computationally much less demanding but no guarantee that the model found will be optimal [17]. Stepwise procedures attempt to identify nonzero $\beta$‘s in equation (2.1). These methods are available in most statistical software packages which make them very popular. There are three types of stepwise selection: forward, backward, and both. Forward stepwise selection begins with a model containing no predictors, called the null model. Next, predictors are added one at a time, until a stopping criteria has been reached or all of the predictors are in the model. At each step of the process, the variable that gives the greatest additional improvement to the fit is added to the model. On the other hand, backward stepwise selection begins with the full model containing all $p$ predictors, and then iteratively deletes the least useful predictor, one at a time until a stopping criteria has been reached or no predictors are left. Backward selection requires that the sample size $n$ is larger than the number of potential predictor variables $p$ so that
the full model can be fit. In contrast, forward stepwise can be used even when $n < p$ which makes it the only viable subset method when $p$ is large. Below are procedures for both forward and backward stepwise selection [9].
Forward Stepwise Selection Procedures

1. Let $M_0$ denote the null model, which contains no predictors.

2. For $k = 0, 1, \ldots, p - 1$:
   
   (a) Consider all $p - k$ models that augment the predictors in $M_k$ with one additional predictor.

   (b) Choose the best among these $p - k$ models, and call it $M_{k+1}$. Here best is defined as having smallest RSS or highest $R^2$.

3. Select a single best model from among $M_0, M_1, \ldots, M_p$ using cross-validated prediction error, $C_p, AIC, AIC_c, BIC$, or adjusted $R^2$.

Backward Stepwise Selection Procedures

1. Let $M_p$ denote the full model, which contains all $p$ predictors.

2. For $k = p, p - 1, \ldots, 1$:
   
   (a) Consider all $k$ models that contain all but one of the predictors in $M_k$, for a total of $k - 1$ predictors.

   (b) Choose the best among the $k$ models and call it $M_{k-1}$. Here best is defined as having smallest RSS or highest $R^2$.

3. Select a single best model from among $M_0, M_1, \ldots, M_p$ using cross-validated prediction error, $C_p, AIC, AIC_c, BIC$, or adjusted $R^2$.

Both forward and backward stepwise selection are not guaranteed to yield the best model containing a subset of $p$ predictors, and both approaches generally produce similar but not identical models. Another approach is a hybrid of both forward and backward stepwise methods, in which variables are added to the model sequentially, and after
adding a new variable, the method may also remove any variables that no longer provide an improvement in the fitted model. Such an approach attempts to mimic best subset selection while retaining the computational advantage of forward and backward stepwise selection [9].

2.4 Penalized Regression

This method of variable selection involves fitting a model using all \( p \) predictors and then the estimated coefficients are shrunken towards zero (relative to least squares estimated coefficients). Shrinkage has the effect of reducing variance in parameter estimates and depending on the type of shrinkage method performed, some coefficients are estimated to be exactly zero. \textit{Ridge regression} and \textit{lasso} are two of the best-known techniques for the shrinkage approach for model selection [9]. \textit{Ridge regression} solves the \( l_2 \)-penalized regression problem of finding \( \beta_0, \beta_1, \ldots, \beta_p \) to minimize

\[
\sum_{j=1}^{p} \beta_j^2 + \lambda \sum_{j=1}^{p} \beta_j^2 = 1
\]  

This is equivalent to minimizing the sum of squares with a constraint of the form \( \sum \beta_j^2 \). The term \( \lambda \sum_{j=1}^{p} \beta_j^2 \) is called the \textit{shrinkage penalty}, is small when \( \beta_0, \beta_1, \ldots, \beta_p \) are closed to zero, and so it has the effect of shrinking the estimates of \( \beta_j \) toward zero. Ridge regression’s advantage over least squares is rooted in the \textit{bias-variance trade-off}. The trade-off is, as the turning parameter \( \lambda \) increases, variance decreases, and bias increases [9]. Unlike best subsets, forward stepwise, and backward stepwise selection, which all generally select models that involve just a subset of the variables, ridge regression will include all \( p \) predictors in the final model. The penalty \( \sum_{j=1}^{p} \beta_j^2 \) in (2.12) will shrink all of the coefficients toward zero, but it will not set any of them exactly to zero (unless \( \lambda = \infty \)). This may not be a problem for prediction accuracy, but it can create a
challenge in model parsimony and interpretation. It can also be troublesome in settings where collecting additional predictor variables can be quite expensive. However, lasso coefficients, \( \hat{\beta}_j^L \), minimize the quantity

\[
RSS + \lambda \sum_{j=1}^{p} |\beta_j|.
\]

(2.15)

Note that as \( \lambda \) changes the number of variables with coefficients different from 0 changes. \( K \) folds cross validation is commonly used to select \( \lambda \). That is, the clean data set is split into \( K \) folds, \( F_1, F_2, \ldots, F_k \), of equal size, then the model is fitted \( K \) times, with each time omitting one of the folds. Then the outcome of this estimation is cross validated by comparing the actual measurements with their prediction for all data in the omitted fold

\[
e_i(\lambda) = \sum_{j \in F_i} y_j - \hat{y}_j(\lambda)^2
\]

(2.16)

The cross validation (CV) error is the error averaged over the \( K \) folds and it is given by

\[
CV(\lambda) = \frac{1}{n} \sum_{i=1}^{k} e_i(\lambda)
\]

(2.17)

When \( K = n \), we called this the leave-one-out cross validation. Conventionally, one can choose \( \lambda \) such that the cross validation error is the minimum [14].

As with ridge regression, the lasso shrinks the coefficient estimates toward zero. However, in the case of lasso, the \( l_1 \) penalty, \( \sum |\beta_j| \) has the effect of forcing some of the coefficient estimates to be exactly equal to zero when the turning parameter \( \lambda \) is large. Lasso has a major advantage over ridge regression because it produces simpler and more interpretable models that involve only a subset of the predictors. In general, lasso performs better in settings where a relatively small number of predictors have substantial coefficients, and the remaining predictors have coefficients that are very small or equal to zero [19]. Ridge regression will perform better when the response is a function
of many predictors, all with coefficients of roughly equal size. However, the number of predictors that is related to the response is never known \textit{a priori} for real data sets. A technique such as cross-validation can be used in order to determine which approach is better on a particular data set. When the least squares estimates have excessively high variance, the lasso solution can yield a reduction in variance at the expense of a small increase in bias, and consequently can generate more accurate predictions. Unlike ridge regression, the lasso performs variable selection, and hence results in models that are easier to interpret [9].

Lasso regression has three major limitations. First, when $p > n$, the lasso selects at most $n$ variables before it saturates. Second, if there is a group of variables among which the pairwise correlations are very high, then the lasso tends to select only one variable from the group and does not consider the other variable to be important. Third, for $n > p$, if there are high correlations between predictors, it has been empirically observed that the predictive performance of the lasso is dominated by ridge regression [23]. As a shrinkage regression method, ridge regression achieves better predictive performance through the bias-variance trade-off. However, ridge regression cannot produce a parsimonious model, for it always keeps all the predictors in the model. Elastic net is a hybrid method of the ridge and lasso regressions. Similar to lasso, the elastic net simultaneously does automatic variable selection and it can select groups of correlated variables. It is like a stretchable fishing net that retains “all the big fish” [23].

2.5 Dimensionality Reduction

Dimensionality reduction, also known as dimension reduction, it is the process of reducing the number of variables in a predictor matrix $p$ to a much smaller transformed,
uncorrelated components $M$ ($p > M$). This is achieved by first compute $M$ different linear combinations of the variables. Then, a least squares model using the $M$ transformed predictor variables is fitted. That is, let $Z_1, Z_2, \ldots, Z_M$ represent $M < p$ linear combinations of our original $p$ predictors, where

$$Z_m = \sum_{j=1}^{p} \phi_{jm} X_j$$

(2.18)

for some constants $\phi_{1m}, \phi_{2m}, \ldots, \phi_{pm}, m = 1, 2, \ldots, M$. One can then fit the linear regression model

$$y_i = \theta_0 + \sum_{m=1}^{M} \theta_m z_{im} + \epsilon_i, \quad i = 1, 2, \ldots, n,$$

(2.19)

using the least squares approach. This method reduced the number of coefficients estimating from $p + 1$ to $M + 1$. If the constants $\phi_{1m}, \phi_{2m}, \ldots, \phi_{pm}$ are chosen wisely, then such dimension reduction approaches can often outperform least squares regression. All dimension reduction methods work in two steps. However, the choice of $Z_1, Z_2, \ldots, Z_M$ can be achieved in different ways. There are two popular approaches: principal components and partial least squares [9]. We will omit the discussion of partial least squares, for details, refer to [9].

### 2.6 Principal Components

There are two steps in constructing principal components. First is Principal Components Analysis (PCA), which is a technique for reducing the dimension of the $n \times p$ data matrix $X$. This transformation is defined in such a way that the first principal component has the largest possible variance, and each succeeding components in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. These uncorrelated orthogonal components are denoted $Z_1, Z_2, \ldots, Z_M$. In general, one
can construct up to \( p \) distinct principal components. Second, once the \( M \) principal components are obtained, we then use these components as predictors in a linear regression model using the least squares method, called *Principal Components Regression* (PCR). The underlying assumption of PCR is that a small number of principal components are sufficient to explain most of the variability in the data, as well as the relationship with the response. In other words, we assume that the directions in which the \( M \) components show the most variation are the directions that are associated with \( Y \). If the assumption holds, then fitting a least squares model to the \( M \) principal components will lead to better results than fitting a least square model with all \( p \) predictors, since most or all of the the information in the data that relates to the response is contained in \( Z_1, Z_2, \ldots, Z_M \), and by estimating only \( M \leq p \) coefficients we can help mitigate overfitting [9]. However, the components can become somewhat hard to interpret in this framework which can limit their usefulness in applications.

### 2.7 Decision Trees

Decision trees can be applied to both regression and classification problems. We will focus on regression problems since our application problem has a continuous outcome. A decision tree is comprised of nodes and splits to the data. The tree starts with all training data in the first node, and an initial split is made using a predictor variable, splitting the data into two child nodes. Splits can then be made from the child nodes. A terminal node is one where no more splits are made. We can continue to use this recursive binary splitting to grow a large tree on the training data, stopping only when each terminal node has fewer than some minimum number of observations. Decision tree has many advantages. One important advantage is the ease of interpretation. While
a tree can be complex, involving many splits and nodes, one can interpret the model. Additionally, making model prediction does not involve calculation; the predictions are based on decision rules [9]. A tree with too many splits may mean over fitting the training data leading to poor generalization in out of sample test data, while a tree with too few of splits may miss out on accuracy. Thus, it is important to use cross validation to determine the size of a decision tree.

Figure 2.1 is an example of a decision tree with five terminal nodes. This tree is pruned from a full tree with 13 terminal nodes. As an illustrative interpretation, we can concluded that the violent crime for communities with percentage of kids born with two parents less than 55.39\% and the percentage people using public transportation is less than 15.935\%, is 1733.0 per 100,000 of the population.

2.8 Bagging, Random Forests, Boosting

Decision trees as described above can suffer from high variance. Bagging, random forests, and boosting use trees as building blocks to construct more powerful prediction models.

Bagging (or bootstrap aggregation) is a procedure for reducing the variance of a statistical learning method. Recall that given a set of $n$ independent observations $Z_1, Z_2, \ldots, Z_n$, each with variance $\sigma^2$, the variance of the mean $\bar{Z}$ of the observations is given by $\sigma^2/n$. In other words, averaging a set of observations reduces variance. To reduce the variance and hence increase the prediction accuracy, bagging methods generate $B$ different bootstrapped training data sets, build a separate prediction model using each bootstrapped training set, and average the resulting predictions. To apply bagging to regression trees, we simply construct $B$ regression tree using $B$ bootstrapped training sets, and average
the resulting predictions. Each individual tree has high variance, but low bias. Averaging these $B$ trees reduces the variance. Overall, bagging improved accuracy over prediction using a single tree, however, it can be difficult to interpret the resulting model. Thus, bagging improves prediction accuracy at the expense of interpretability [9].

*Random forests* provide an improvement over bagged trees by way of *de-correlating* the trees. As in bagging, we build a number of decision trees on bootstrapped training samples. However, when building these decision trees, each time a split in a tree is considered, a random sample of the $m$ predictors is chosen as split candidates from the full set of $p$ predictors. The split is allowed to use only one of the $m$ predictors. A fresh
sample of $m$ predictors is taken at each split and typically we choose $m \approx \sqrt{p}$ [9].

Boosting is another approach for improving the predictions resulting from a decision tree. Recall that bagging involves creating multiple copies of the original training data set using the bootstrap, fitting a separate decision tree to each copy, and then combining all of the trees in order to create a single predictive model. Boosting works in a similar way, except that the trees are grown sequentially: each tree is grown using information from previously grown trees. Boosting does not involve bootstrap sampling; instead each tree is fit on a modified version of the original data set. Note that unlike in bagging, the construction of each tree depends strongly on the trees that have already been grown [9].

2.9 Discussion of Methods

There are inconclusive results in the review of model selection methods in the current literature. In a study [13], Murtaugh concludes that stepwise methods performed similarly to exhaustive algorithms for selections, and the choice of criterion for comparing models had little effect on predictive ability. The author further argued that there is no best method of variable selection and that any regression based approaches are capable of yielding useful predictive models [13]. In a comparison study of AIC and BIC using data in selection of stock-recruitment (SR) relationships using likelihood ratio tests, Wang found that both AIC and BIC are valid in selecting the most suitable SR relationships, and BIC is a better criterion than AIC for nested models [21]. However, Burnham argued that BIC is a poor criterion and suggested that it should not be used with real data [5]. Many studies suggested that the use of stepwise regression is a bad practice and it is very poor procedure. Stepwise regression has three main drawbacks and they are: bias in parameter estimation, inconsistencies among model selection algorithms, and
inappropriate focus or reliance on a single best model [22, 10, 5]. Many articles in current statistical literature found that null hypothesis test (NHT) based algorithms such as stepwise selection has been over used and misused, which adds little value to the products of research. Many proposed that information theoretic (IT) analysis is a superior alternative to NHT [22, 10, 5].
Chapter 3

Model Selection Results

In this section, we present the results of the following model selection strategies applied to our crime database: forward, backward, and both forward and backward simultaneously (we will simply called it bothFB), lasso, elastic net, count, best random subset selection (BRSS), decision tree, tree with pruning, bagging, and random forest. Our goal here is to select the “best” model for prediction and draw inference about what set of socio-demographic variables are most associated with crime. We used several different model selection criteria including: multiple r-squared ($R^2$), adjusted r-squared ($R^2_a$), AIC, mean squared error (MSE), and whether a model is parsimonious. There are three key parts in this chapter. One is to introduce two new model selection methods, count and BRSS. Two is to split the complete database into training and test sets and fit models: forward, backward, bothFB, lasso, elastic net, BRSS, and count. We will also fit a decision tree, tree with pruning, bagging, random forest, and principal component models on the training data set. Then we performed cross-validation of these models on the test data using multiple r-squared ($R^2$), adjusted r-squared ($R^2_a$), mean squared error (MSE) and AIC. To compare model performance, we fit each one of the models 1,000 times.
with different partitions of training and test data splits.

### 3.1 Best Random Subset Selection

The goal of our best random subset selection model is to approximate best subset selection without overfitting the training data and without having the computational expense of carrying out the full best subsets algorithm. In the cleaned data set, there are 94 predictors which means we have $2^{94}$ possible models. When the number of predictors $p$ is large, the best subset selection is computationally intensive, and so BRSS is needed. Let $p$ be the number of predictors in a model and $N$ be the number of randomly selected models with $p$ predictors. For each $p$, as $N$ approaches infinity, we successfully approximate best subset selection. If we limit the maximum number of predictors in our models $p$ to $Q$, where $Q$ is much less than 94, we can reduce bias and the risk of overfitting our data. We will use the plot of $p$ versus training $R^2_a$ to help us determine the number of predictors we should use in the BRSS model.

**Procedure to determine $p$ in BRSS model**

1. Randomly split the cleaned data into 75% training and 25% test sets. That leaves 605 observations in the training set and 202 observations in the test set.

2. Fit $N$ models of $p$ predictors on the training data set and pick the model with the maximum $R^2_a$ for each $p$, where $p = 1, 2, 3, \ldots, 94$ and where $N$ will typically be, say, 100 or 1,000. Later, we will show how BRSS model perform for $N = 10, 50, 100, 500, 1000, 5000, 10000, 50000$ and 100000.

3. For each of the 94 best of the $N$ models with $p$ predictors, we predict violent crime in the test data set and obtain the test $R^2_a$. 

34
4. Plot both training and test $R^2_a$ versus $p$. We selected $Q$ such that it is the optimal upper bound of $p$ based on the plots, and we limited $p < Q$.

As an illustrative example, suppose we let $N = 100$, meaning we would train 100 randomly chosen models with one predictor, train 100 randomly chosen models with two predictors, train 100 randomly chosen models with three predictors, and etc. Then, for each of the 100 models, select the model with the maximum $R^2_a$ and make prediction of violent crime in the test data set and obtain the test $R^2_a$. In this setting we would fit a total of 9,400 models (94*100). $Q$ is much smaller than $p$ (the plots of training and test $R^2_a$ versus $p$ illustrates the optimal $Q$). We will show how the BRSS model performs for $N = 10, 50, 100, 500, 1000, 5000, 10000, 50000$ and 100000.

Table 3.1 shows the distribution of the results of $R^2_a$ for the training BRSS models and test BRSS models on all 94 predictors, where $p$ is the number of predictors in the best of the corresponding $N$ models. Note that the number of predictors in the best training models range from 64 to 87 predictors, where the number of predictors in the model that yield the best test $R^2_a$ can be as low as 21. The best training model has an $R^2_a$ of 0.7211 and 77 predictors. This is the model obtained from $N = 100,000$. The overall best model base on cross validation has an $R^2_a$ of 0.627. This is the model selected from $N = 50$ with 32 predictors in the training model.

Figure 3.1 are plots of training and test $R^2_a$ on $p$ where $p = 1, 2, \ldots, 94$, the number of predictors in a model for $N = 10, 50, 100$ and 500. The x-axis is the number of predictors in a model and the y-axis is $R^2_a$. For $p < 20$ and $N$ from 10 to 50, there is a drastic increase in $R^2_a$ and a small difference between training and test $R^2_a$. As $N$ increases, the training $R^2_a$ increase toward the mean of the $N$ training $R^2_a$, and continue to increase slightly afterward. However, this does not translate to test $R^2_a$. Test $R^2_a$ form a trend of increasing followed by slight decease as the number of predictors $p$ increases in a model.
The discrepancy between training $R^2_a$ and test $R^2_a$ increases as the number of predictors approach 94. This discrepancy is due to training model overfitting the training data and as a training model gets more complex, the error in predicting out of sample increases which is expected.

Figure 3.2 are plots of training and test $R^2_a$ on $p$ for $N = 1000, 5000, 10000$ and 100000. There are no observable differences between these plots for different values of $N$. For $p$ less than about 15, there are significant increases in the training and test $R^2_a$, and minimum discrepancy between training and test $R^2_a$. Based on both Figures 3.2 and 3.3, we can conclude that for $N = 1,000$, including no more than 25 ($p < 25$) predictors
in our training model can explain vast majority of the variance in crime rate from our training model and minimize the discrepancy between training and test $R^2_A$. Note that increasing $N$ after 1,000 only increase computation time but we see no difference in results. In other words, we are comfortable using $N = 1,000$ and $p = 20$ in the best random subset selection model.
Figure 3.2: Best Random Subset: Adjusted R-squareds vs. the number of predictors in BRSS models

- The x-axis, $p$, is the number of predictors in a model
- The y-axis, $\text{aR}^2$, is the adjusted r-squared
- The red circles are training adjusted r-squared, and the blue squares are test adjusted r-squared. Horizontal lines are the mean of adjusted r-squared
Best Random Subset Procedures for $N = 1,000$ and $p = 20$

1. Randomly split cleaned data into 75% training and 25% test sets.

2. For each value of $i = 1, 2, \ldots, 20$, fit $N = 1,000$ models with $i$ predictors on the training data set and pick the model with the maximum $R^2_a$ for each $i$. At the end of this step, there are 20 best models, one for each value of $i$.

3. Among the best models $B_1, B_2, \ldots, B_{20}$, pick the best using training $R^2_a$ of the model.

4. Cross validate the selected overall best model using the test data set.

There are $2^{94}$ possible models, using $N = 1,000$ and $p = 20$, BRSS systematically, and randomly fit a total of 20,000 models. See Procedure to Determine $p$ in BRSS model for details on how to pick $N$ and $p$ that minimize over fitting and maintain model simplicity ($p < Q$).

Table 3.2 shows the results of BRSS model with $N = 1,000$ and $p = 20$. The BRSS model selected 15 predictors with training and test $R^2_a$ of 0.6836 and 0.6486, respectively. Note that for any $N$ greater than 100 and $p$ around 20 would yield good results as well (see Table 3.3).

Table 3.3 summarizes the distributions of various BRSS models with $N = 200$ and $p = 8, 15, 20$ and 24. $M$ is the number of times the complete data set splits into 75% and 25% test sets, $p$ is the maximum number of predictors in a model, and $N$ is the number of models we train for each $p$. As indicated in Table 3.3, with only up to 8 predictors in a model and fit 200 models of each $p$, the mean training and test $R^2_a$ of fitting BRSS model 8 times are 0.648 and 0.6238, respectively. These are desirable results with more parsimonious models.
Table 3.2: BRSS with $N = 1000$ and $p = 20$

| Predictors                        | Estimate | Std. Error | t value | Pr(>|t|) |
|-----------------------------------|----------|------------|---------|----------|
| (Intercept)                       | -8091    | 3495       | -2.315  | 0.020953 |
| PctRecImmig5                      | -64.97   | 18.27      | -3.557  | 0.000406 |
| racepctblack                      | 6.762    | 1.81       | 3.736   | 0.000205 |
| agePct65up                        | 21.28    | 5.752      | 3.7     | 0.000235 |
| PctEmploy                         | 4.878    | 3.527      | 1.383   | 0.167173 |
| PctForeignBorn                    | 24.93    | 5.283      | 4.719   | 2.96E-06 |
| PctPersOwnOccup                   | 0.4028   | 2.041      | 0.197   | 0.843662 |
| NumKidsBornNeverMar               | 0.0068   | 0.00196    | 3.453   | 0.000595 |
| MedYrHousBuilt                    | 3.922    | 1.768      | 2.218   | 0.02691  |
| PctPersDenseHous                  | 24.15    | 10.61      | 2.276   | 0.023177 |
| PctLess9thGrade                   | -13.28   | 5.117      | -2.596  | 0.009665 |
| MedOwnCostPctIncNoMtg             | -38.17   | 12.52      | -3.049  | 0.002399 |
| PctHousNoPhone                    | 30.94    | 7.571      | 4.087   | 4.98E-05 |
| PctLargHouseFam                   | -6.362   | 12.28      | -0.518  | 0.604464 |
| PctKidsBornNeverMar               | 96.95    | 10.83      | 8.954   | <2e-16   |
| MalePctDivorce                    | 37.94    | 8.042      | 4.717   | 2.99E-06 |

Number of Variables: 15
Training aR2: 0.6836
Training R2: 0.6914
Training AIC: 8876.3
Test aR2: 0.6486
Test R2: 0.6749
Test AIC: 3030.49

3.2 Count and Others

Count 4 is a consensus model; this model is constructed using predictors that are selected at least four times by six of the different models: forward, backward, bothFB, lasso, elastic net and BRSS. Similarly, count 5 is constructed using predictors that are selected at least five times by other models. The elastic net model was with $\alpha = 0.5$, a representative midpoint between ridge and lasso.

Table 3.4 and Table 3.5 show the results of fitting each of the models once on the
Table 3.3: BRSS Models with $N = 200$ and Varying $p$

<table>
<thead>
<tr>
<th>$M=10$, $N=200$, $p=24$</th>
<th>Min.</th>
<th>1st Qu.</th>
<th>Median</th>
<th>Mean</th>
<th>3rd Qu.</th>
<th>Max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>train ar$^2$</td>
<td>0.6481</td>
<td>0.6587</td>
<td>0.5718</td>
<td>0.6705</td>
<td>0.6818</td>
<td>0.6958</td>
</tr>
<tr>
<td>test ar$^2$</td>
<td>0.6018</td>
<td>0.6219</td>
<td>0.6496</td>
<td>0.6554</td>
<td>0.6906</td>
<td>0.7104</td>
</tr>
<tr>
<td>train MSE</td>
<td>126000</td>
<td>134600</td>
<td>139200</td>
<td>138700</td>
<td>142600</td>
<td>151900</td>
</tr>
<tr>
<td>test MSE</td>
<td>140500</td>
<td>157700</td>
<td>171900</td>
<td>177100</td>
<td>198500</td>
<td>225000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$M=10$, $N=200$, $p=20$</th>
<th>Min.</th>
<th>1st Qu.</th>
<th>Median</th>
<th>Mean</th>
<th>3rd Qu.</th>
<th>Max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>train ar$^2$</td>
<td>0.6559</td>
<td>0.6673</td>
<td>0.6739</td>
<td>0.674</td>
<td>0.6837</td>
<td>0.6913</td>
</tr>
<tr>
<td>test ar$^2$</td>
<td>0.6009</td>
<td>0.6351</td>
<td>0.6463</td>
<td>0.6466</td>
<td>0.6571</td>
<td>0.6921</td>
</tr>
<tr>
<td>train MSE</td>
<td>128400</td>
<td>134400</td>
<td>138300</td>
<td>138200</td>
<td>142600</td>
<td>145600</td>
</tr>
<tr>
<td>test MSE</td>
<td>142500</td>
<td>156100</td>
<td>187000</td>
<td>179800</td>
<td>193900</td>
<td>232300</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$M=10$, $N=200$, $p=15$</th>
<th>Min.</th>
<th>1st Qu.</th>
<th>Median</th>
<th>Mean</th>
<th>3rd Qu.</th>
<th>Max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>train ar$^2$</td>
<td>0.6516</td>
<td>0.658</td>
<td>0.663</td>
<td>0.6651</td>
<td>0.671</td>
<td>0.6841</td>
</tr>
<tr>
<td>test ar$^2$</td>
<td>0.5852</td>
<td>0.6302</td>
<td>0.6525</td>
<td>0.6443</td>
<td>0.6645</td>
<td>0.6835</td>
</tr>
<tr>
<td>train MSE</td>
<td>141600</td>
<td>143600</td>
<td>149100</td>
<td>148000</td>
<td>149600</td>
<td>158500</td>
</tr>
<tr>
<td>test MSE</td>
<td>134500</td>
<td>145400</td>
<td>155300</td>
<td>155900</td>
<td>167700</td>
<td>179000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$M=10$, $N=200$, $p=8$</th>
<th>Min.</th>
<th>1st Qu.</th>
<th>Median</th>
<th>Mean</th>
<th>3rd Qu.</th>
<th>Max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>train ar$^2$</td>
<td>0.6209</td>
<td>0.6375</td>
<td>0.6476</td>
<td>0.6482</td>
<td>0.6564</td>
<td>0.6793</td>
</tr>
<tr>
<td>test ar$^2$</td>
<td>0.5685</td>
<td>0.5976</td>
<td>0.6321</td>
<td>0.6238</td>
<td>0.6397</td>
<td>0.6762</td>
</tr>
<tr>
<td>train MSE</td>
<td>129200</td>
<td>150400</td>
<td>154700</td>
<td>152400</td>
<td>159300</td>
<td>162600</td>
</tr>
<tr>
<td>test MSE</td>
<td>141800</td>
<td>171200</td>
<td>180500</td>
<td>179400</td>
<td>192200</td>
<td>215800</td>
</tr>
</tbody>
</table>

- $M$ is the number of times we split the data set into training and test set.
- $N$ is the number of models we train for each $p$.
- $p$ is the maximum number of predictors in a brst model.

training data set and then assessing their fit on the test data set. There are 8 predictors that are selected by at least five of these models and they are: percentage of population that is African American, percentage of households with retirement income, percentage of people 16 and over who are employed in manufacturing, percentage of people 16 and over who are employed in professional services, percentage of kids in family housing.
with two parents, percentage of kids born to never married parents, percentage of persons in dense housing, and median owners cost of rent as a percentage of household income (for owners without a mortgage).

As indicated by Table 3.5, stepwise backward selection model has the highest $R^2_a$ of 0.7281 on the training data, with 39 predictors but lowest test $R^2_a$. Thus, the backward model has a high risk of overfitting. The count 4 model has the highest test $R^2_a$ of 0.6697, with only 15 predictors. Model BRSS performs very similarly to count 4, with the same number of predictors and test $R^2_a$ of 0.6486. Count 5 has only 8 predictors in the model, which is the lowest in this group, with test $R^2_a$ of 0.6465 and training $R^2_a$ of 0.6547. For this group of models: forward, backward, bothFB, lasso, and elastic net, the higher the number of predictors in the model led to lower test $R^2$. Thus, these models were overfitted in the training data and did not generalize as well to the test data. Lasso and elastic net selected similar predictors with lasso selected one extra variable. Models BRSS, count 4, and count 5 all have a small discrepancy between training and test $R^2$ and a low number of predictors in the training model meaning they have less biased measures of fit and produce more parsimonious models which are both desirable properties.
Table 3.4: Summary of fitting and cross validating each of the models on one split of the cleaned data into training and test sets (part 1).

<table>
<thead>
<tr>
<th>Predictors</th>
<th>Forward</th>
<th>Backward</th>
<th>bothFB</th>
<th>lasso</th>
<th>Elastic</th>
<th>B RSS</th>
<th>Count4</th>
<th>Count5</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>household size</td>
<td>-247.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>racePctWhite</td>
<td>2.158</td>
<td>7.069</td>
<td>-0.213</td>
<td>-1.352</td>
<td>4.724</td>
<td></td>
<td></td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>racePctAsian</td>
<td>10.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>racePctHisp</td>
<td>4.945</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>agePct12to19</td>
<td></td>
<td>-4.372</td>
<td>-4.065</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>agePct55up</td>
<td>21.28</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>pctWWeekly</td>
<td>-14.21</td>
<td>-20.03</td>
<td>-20.44</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>pctlWInvREc</td>
<td>-7.73</td>
<td></td>
<td>-1.631</td>
<td>-1.685</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>whitePerCap</td>
<td>-0.0171</td>
<td></td>
<td>-0.016</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>blackPerCap</td>
<td>-0.0054</td>
<td></td>
<td>-0.001</td>
<td>-0.002</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>indianPerCap</td>
<td>0.0002</td>
<td>0.0002</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>OtherPerCap</td>
<td>0.0005</td>
<td>0.0004</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>HispPerCap</td>
<td>0.0015</td>
<td>0.0017</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>NumUnderPov</td>
<td>0.0064</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>FctlPoPUnderPov</td>
<td>-9.668</td>
<td>-19.56</td>
<td>-16.33</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>PctlTest5thGrade</td>
<td>-14.38</td>
<td></td>
<td>-13.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Pctl35orMore</td>
<td>8.744</td>
<td>14.11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>PctlUnemployed</td>
<td>-14.63</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>PctlEmployn μ</td>
<td>-5.928</td>
<td>-5.312</td>
<td>-7.905</td>
<td>-1.396</td>
<td>-1.218</td>
<td>-6.866</td>
<td>-10.53</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>PctlEmployn ProfServ</td>
<td>-5.636</td>
<td>-10.2</td>
<td>-12.02</td>
<td>-1.329</td>
<td>-1.310</td>
<td>-6.353</td>
<td>-10.72</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>MalePctDivorce</td>
<td>100.7</td>
<td></td>
<td>7.988</td>
<td>9.022</td>
<td>37.94</td>
<td>27.38</td>
<td></td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>FemalePctDivorce</td>
<td>-101.4</td>
<td></td>
<td>-121.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>TotalPctDiv</td>
<td>106</td>
<td>-98.14</td>
<td>136.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>PctlYoungKids2Par</td>
<td></td>
<td>6.75</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>PctlTeen12Par</td>
<td>-5.442</td>
<td></td>
<td>-2.110</td>
<td>-3.943</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>NumKidsBornNeverMar</td>
<td>0.0106</td>
<td>0.0014</td>
<td>0.0021</td>
<td>0.007</td>
<td>-0.004</td>
<td></td>
<td></td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>FctlKidsBornNeverMar</td>
<td>47.38</td>
<td>39.08</td>
<td>53.11</td>
<td>64.11</td>
<td>58.85</td>
<td>96.95</td>
<td>64.78</td>
<td>66.91</td>
<td>6</td>
</tr>
<tr>
<td>NumImmigRecent</td>
<td>-0.0039</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>PctlAppRecent</td>
<td>7.576</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>PctlAppRec5</td>
<td>-8.857</td>
<td>-1.235</td>
<td>-1.139</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>PctlAppRecimmig5</td>
<td></td>
<td>-65</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>PctlAppRecimmig10</td>
<td>-34.81</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>PctlAppImmig10Only</td>
<td>-4.826</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>PctlAppImmig10</td>
<td></td>
<td>-6.36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>PctlAppHouseOccup</td>
<td>-38.87</td>
<td>-56.15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>PersPerOccHous</td>
<td>206.2</td>
<td>1332</td>
<td>766.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>PersPerRentOccHous</td>
<td>-938.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>PctlPersOwnOccp</td>
<td>-56.09</td>
<td>0.403</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
</tbody>
</table>

43
Table 3.5: Summary of fitting and cross validating each of the models on one split of the cleaned data into training and test sets (part 2).

<table>
<thead>
<tr>
<th>Predictors</th>
<th>Forward</th>
<th>Backward</th>
<th>bothFB</th>
<th>lasso</th>
<th>Elastic</th>
<th>BRSS</th>
<th>Count4</th>
<th>Count5</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>PctHousLess3BR</td>
<td>5.55</td>
<td>1.190</td>
<td>1.14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>HousVacant</td>
<td>0.0159</td>
<td>0.0171</td>
<td>0.0124</td>
<td>0.0114</td>
<td>0.0147</td>
<td></td>
<td></td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>PctHousOcc</td>
<td>-3.477</td>
<td>-4.806</td>
<td>-5.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>PctHousOwnOcc</td>
<td>53.86</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>PctVacantBoarded</td>
<td>11.15</td>
<td>11.78</td>
<td>7.748</td>
<td>8.601</td>
<td>6.327</td>
<td></td>
<td></td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>PctVacMore6Mos</td>
<td>3.227</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>MedYrHousBuilt</td>
<td>0.0639</td>
<td></td>
<td>3.922</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>PctHousNoPhone</td>
<td>0.5028</td>
<td>1.120</td>
<td>30.94</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>OwnOccLowQuart</td>
<td>-0.0094</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>OwnOccMedVal</td>
<td>0.0100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>OwnOccHiQuart</td>
<td>-0.0023</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>RentLowQ</td>
<td>-2.148</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>RentMedian</td>
<td>2.42</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>MedOwnCostPctInc</td>
<td>-19.72</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>MedOwnCostPctIncNoMtg</td>
<td>-70.13</td>
<td>-53.04</td>
<td>-64.07</td>
<td>-37.35</td>
<td>-33.99</td>
<td>-38.17</td>
<td>-42.5</td>
<td>-42.15</td>
<td>6</td>
</tr>
<tr>
<td>PctForeignBorn</td>
<td>19.25</td>
<td>2.341</td>
<td>2.036</td>
<td>24.93</td>
<td>8.862</td>
<td></td>
<td></td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>PctBornSameState</td>
<td>-2.007</td>
<td>-1.462</td>
<td>-1.419</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>PctSameHouse85</td>
<td>-6.764</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

Number of Variables: 21  30  23  28  27  15  15  8

Train R2: 0.7092  0.7281  0.7154  0.652  0.659  0.6836  0.6738  0.6547
Train R2: 0.7193  0.7456  0.7263  0.6526  0.6595  0.6914  0.6819  0.6503
Train AIC: 8881  8857  8870  8970  8958  8876  8895  8922
Test R2: 0.5959  0.5234  0.527  0.5545  0.5634  0.6486  0.6697  0.6465
Test R2: 0.5979  0.5258  0.5947  0.5567  0.5655  0.6749  0.6944  0.6606
Test AIC: 2996  3030  2998  3016  3012  3030  3018  3025
3.3 Regression Tree

In this section, we use the class of decision trees including: tree with pruning, bagging, and random forest to make predictions. Again, we split the cleaned data into 75% training and 25% test sets and fit each of the models on the training data, and assess the fit of the training models on the test data.

Figure 3.3 is a decision tree without pruning on the training data. The tree has 12 terminal nodes and includes 9 predictors. These predictors are population for community, percent of people using public transit for commuting, percentage of households with investment/rent income, percentage of people 16 and over in the labor force and unemployed, percentage of kids in family housing with two parents, number of kids born to never married parents, percent of people who speak only English, and percent of vacant housing that is boarded up.

Figure 3.3: Full Decision Tree

* Full tree has 12 terminal nodes and 9 predictors.
* Predicted variable is ViolentCrimePerPop, it is the total number of violent crimes per 100K population
* Mean ViolentCrimePerPop is 770.3

Figure 3.4 is a scree plot of the size of a tree against deviance. The size of a tree
is the number of terminal nodes on a tree. As indicated by the plot, a tree with four or five terminal nodes minimizes the deviance. Next, we prune the full tree to five terminal nodes.

Figure 3.5 is a tree with pruning with five terminal nodes. One advantage of decision tree lies in simple interpretations with no confusing coefficients for the investigator to interpret. For example, in communities with the percent of people using public transit for commuting less than 15.94% and the percentage of kids in family housing with two parents is less than 55.39%, the predicted violent crime rate is 1733 per 100,000 population.

Figure 3.6 are the results of fitting the full decision tree, tree with pruning, bagging, and random forest to the training data, with each model then applied to the test data. Note that the advantage of decision trees and trees with pruning lies in simplicity of interpretation and drawing inference. With bagging and random forest, drawing inference
is difficult. Thus, the goal of bagging and random forest should be utilized for making accurate predictions. The tree built with pruning has a test $R^2$ of 0.413 which is much higher than the test $R^2$ in full tree but with less than half the number of predictors. Random forest has a test $R^2$ of 0.593, which is slightly higher than bagging.

### 3.4 Principal Components

In addition to using principal components to make predictions as we’ve done with the other methods, another main question we are going to address here is what’s the difference in training and test MSE and $R^2$ when the number of components used in training our model varies? Similar to other regression models, we first fit the model on the training data and cross validate on the test data. Next, we plot the training and test MSE versus the number of components used in principal components regression (PCR) on the
training data to examine the relationship between model complexity and prediction error. We will also plot the training and test $R_a^2$ versus the number of components used in PCR on the training data to illustrate the trade-off between model complexity and predictive performance.

Table 3.7 is a summary of the first ten principal components (PCs). Figure 3.6 is a
scree plot of the first ten components. As indicated by the plot, first four PCs explained a majority (59.6%) of the variance. Thus, a principal component regression with 4 PCs may be optimal.

**Procedures for principal component regression with different PCs**

1. Split the cleaned data set with 75% training and 25% test data sets.

2. Fit principal component regression (PCR) on the training data set with the first $p$ components, where $p = 1, 2, 3, \ldots, 94$.

3. Make prediction using each of the 94 models on test data.

Figure 3.7 is a plot of the training and test error (MSE) by the number of principal components (PCs) used in the model. As indicated by the plot, principal component regression models (PCR) with seven or less components has minimum discrepancy between training and test MSE. After the seventh PC, as the number of PCs increases,
training error continues to decrease but test error increases slightly. Thus, the discrepancy between training and test error increases after the seventh PC. This plot accurately depicts the trade off between model complexity and error.

Figure 3.8 is a plot of training and test $R^2_a$. The x-axis is the number of PCs, and the y-axis is $R^2_a$. As shown in the plot, PCR with up to seven components has minimum discrepancy between training and test $R^2_a$. The result for fitting PCR with the first seven PC is 0.6108 and 0.5575 for training and test $R^2_a$, respectively. After the seventh PC, as the number of PCs increases in the model, the training $R^2_a$ continues to increase whereas test $R^2_a$ decreases slightly. The training $R^2_a$ approaches 0.8 where the test $R^2_a$ bounded below 0.6. Thus, as we found in Figure 3.7, as a model gets more complex, it tends to
overfit the training data and the error in predictions on the test data increases.

### 3.5 Model Performance

Figure 3.9 is a plot of training $R^2$ with test $R^2$. It is a comparison of model performance for fitting each one of the following models once: forward, backward, bothFB, lasso, elastic net, BRSS, count 4 and count 5, decision tree, tree with pruning, bagging, and random forest. The dashed line is plotted at $y = x$ to represent perfect concordance between training and test $R^2$. The training $R^2$ for bagging and random forest are more than
Figure 3.9: Plot of training $R^2$ versus test $R^2$ on one split of the cleaned data for each of the models: forward, backward, both forward and backward simultaneously, lasso, elastic net, best random subset selection, count 4, count 5, principal component regression, decision tree, tree with pruning, bagging, and random forest.

- The dashed line is $y = x$.
- The x-axis is training r-squared.
- The y-axis is test r-squared.
- Model BRSS uses adjusted r-squared.

0.9 but their test $R^2$ are below 0.6 representing extreme overoptimism in fit when they are fitted to external data. The tree without pruning has training $R^2$ close to 0.8 but test $R^2$ only around 0.3 which was the worst performer. Thus, random forest, bagging and tree
models have big discrepancies between training and test $R^2$ creating a high risk of overfitting. Trees with pruning perform a litter better than the full tree, bagging, and random forest, with training $R^2$ around 0.65 and test $R^2$ around 0.4 but still overfit the training data. Since random forest and bagging models have the highest training $R^2$ but lowest test $R^2$ of all models, these models are the “worst” of these group. Stepwise procedures perform better than the regression trees but still have big discrepancies between training and test $R^2$, with training $R^2$ more than 0.7 and test $R^2$ below 0.6. Thus, these groups of models tend to overfit the training data in this example. Elastic net ($\alpha = 0.05$) and lasso perform similarly to each other, and these two models perform slightly better than stepwise models. Principal component regression has training $R^2$ slightly above 0.6 and test r-squared slightly below 0.6 so this model perform a little better than lasso and elastic net. The performance of BRSS, count 4 and count 5 are on the $y = x$ line and all perform significantly better than other models. The training $R^2$ for models BRSS, count 4, and count 5 are almost as high as backward model and perform equally well on test data. In addition, the number of predictors in these models are much smaller compared to lasso, elastic net, and stepwise models. Overall, because model count 4 has only 8 predictors, which is less than half of the other models, with almost no discrepancy between training and test r-squared, count 4 is the winner of this group. Model BRSS performance is only slightly below count 4. Clearly, count 4, count 5 and BRSS methods outperform all current methods both in terms of overall predictive performance and bias of training $R^2$.

However, since we only partitioned the data set once, it is difficult to tell whether one model consistently performs better than others in predicting out of sample test data. Therefore, now we tested these models for consistency in performance by repeating this process one thousand times, each with a different training and test split.
3.6 Simulation Study of 1000 Training and Test Splits

In this section, we split the cleaned data set randomly 1,000 times into training set and test set, and for each time we split the data, fit each of the following methods: forward, backward, bothFB, lasso, elastic net, and BRSS, decision tree, tree with pruning, bagging, and random forest, and then make predictions on the test data set. The goal here is to compare mean overall performance in predictions and find the best model based on multiple r-squared. Note the BRSS model used adjusted $R^2$ as criteria for selecting the best model whereas other models used multiple $R^2$.

Procedure: All Methods 1000 Partitions

1. Split the cleaned data set into training set and test set where 75% of the cleaned data is training set and the remaining 25% is test set. There are $n = 605$ observations in the training set, and $n = 202$ in the test set.

2. With the training data set, fit regression models using stepwise forward, backward, both forward and backward simultaneously, lasso, elastic net, best random subset methods (with $M = 1, N = 1000,$ and $p = 20$), decision tree, tree with pruning, bagging, and random forest. Obtain training $R^2$ for each one of the methods.

3. Use each of the training models to make predictions on the test data, and obtain multiple $R^2$ on the test data set for each of the methods.

4. Repeat step 1 to step 3 one thousand times and find the mean of the multiple $R^2$ for stepwise forward, backward, both forward and backward simultaneously, lasso, elastic net, decision tree, tree with pruning, bagging, random forest, and find the average of $R_a^2$ for the BRSS method.

Table 3.8 is the summary of 1,000 multiple $R^2$ of forward, backward, bothFB, lasso,
Table 3.8: Summary of training r-squared and test r-squared for spitting the cleaned data 1,000 times into 75% training and 25% test sets.

<table>
<thead>
<tr>
<th></th>
<th>Min.</th>
<th>1st Qu.</th>
<th>Median</th>
<th>Mean</th>
<th>3rd Qu.</th>
<th>Max.</th>
<th>sd</th>
</tr>
</thead>
<tbody>
<tr>
<td>train R2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bothFD</td>
<td>0.6531</td>
<td>0.6909</td>
<td>0.7014</td>
<td>0.7013</td>
<td>0.7116</td>
<td>0.7466</td>
<td>0.0153</td>
</tr>
<tr>
<td>forward</td>
<td>0.6535</td>
<td>0.692</td>
<td>0.7029</td>
<td>0.7029</td>
<td>0.7136</td>
<td>0.7457</td>
<td>0.0154</td>
</tr>
<tr>
<td>backward</td>
<td>0.6749</td>
<td>0.7131</td>
<td>0.7225</td>
<td>0.7224</td>
<td>0.732</td>
<td>0.7646</td>
<td>0.0145</td>
</tr>
<tr>
<td>lasso</td>
<td>0.5299</td>
<td>0.6055</td>
<td>0.6214</td>
<td>0.6206</td>
<td>0.6365</td>
<td>0.7065</td>
<td>0.0241</td>
</tr>
<tr>
<td>elastic</td>
<td>0.4941</td>
<td>0.6048</td>
<td>0.6221</td>
<td>0.6202</td>
<td>0.6387</td>
<td>0.6983</td>
<td>0.0267</td>
</tr>
<tr>
<td>BRSS*</td>
<td>0.6171</td>
<td>0.6625</td>
<td>0.6719</td>
<td>0.6712</td>
<td>0.6812</td>
<td>0.7136</td>
<td>0.0142</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Min.</th>
<th>1st Qu.</th>
<th>Median</th>
<th>Mean</th>
<th>3rd Qu.</th>
<th>Max.</th>
<th>sd</th>
</tr>
</thead>
<tbody>
<tr>
<td>test R2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bothFB</td>
<td>0.1177</td>
<td>0.5796</td>
<td>0.6228</td>
<td>0.6115</td>
<td>0.6575</td>
<td>0.7452</td>
<td>0.0678</td>
</tr>
<tr>
<td>forward</td>
<td>0.1893</td>
<td>0.5794</td>
<td>0.6227</td>
<td>0.6121</td>
<td>0.6573</td>
<td>0.7439</td>
<td>0.0666</td>
</tr>
<tr>
<td>backward</td>
<td>0.0285</td>
<td>0.5542</td>
<td>0.6019</td>
<td>0.5891</td>
<td>0.6371</td>
<td>0.7345</td>
<td>0.0733</td>
</tr>
<tr>
<td>lasso</td>
<td>0.4655</td>
<td>0.5796</td>
<td>0.6078</td>
<td>0.6071</td>
<td>0.6368</td>
<td>0.7338</td>
<td>0.0411</td>
</tr>
<tr>
<td>elastic</td>
<td>0.4679</td>
<td>0.5793</td>
<td>0.607</td>
<td>0.6054</td>
<td>0.6331</td>
<td>0.7295</td>
<td>0.0402</td>
</tr>
<tr>
<td>BRSS*</td>
<td>0.5304</td>
<td>0.6295</td>
<td>0.6591</td>
<td>0.6596</td>
<td>0.6925</td>
<td>0.8024</td>
<td>0.0457</td>
</tr>
<tr>
<td>tree</td>
<td>0.2957</td>
<td>0.6093</td>
<td>0.6529</td>
<td>0.6475</td>
<td>0.6951</td>
<td>0.8111</td>
<td>0.0665</td>
</tr>
<tr>
<td>prune</td>
<td>0.3171</td>
<td>0.5245</td>
<td>0.5594</td>
<td>0.5617</td>
<td>0.6086</td>
<td>0.7132</td>
<td>0.0670</td>
</tr>
<tr>
<td>bag</td>
<td>0.4276</td>
<td>0.6158</td>
<td>0.6477</td>
<td>0.6427</td>
<td>0.6726</td>
<td>0.7597</td>
<td>0.0448</td>
</tr>
<tr>
<td>rf</td>
<td>0.5164</td>
<td>0.6364</td>
<td>0.6610</td>
<td>0.6582</td>
<td>0.6842</td>
<td>0.7610</td>
<td>0.0380</td>
</tr>
</tbody>
</table>

* BRSS model uses adjusted r-squared

elastic net, decision tree, tree with pruning, bagging and random forest for both the training and test sets. For model BRSS, this is the mean of 1,000 adjusted $R^2$, in addition, model BRSS has no more than 20 predictors in the training models as defined earlier. The BRSS method had the highest mean test $R^2$ and so on average it out-performed all the methods in this group. The random forest test $R^2$ is slightly lower than BRSS model, second highest of the group, but suffered from severe over-optimism with significantly higher average training $R^2$. On average, regression tree methods have high mean training
$R^2$ and low test $R^2$. Stepwise models perform slightly better than regression tree models in terms of overfitting.

Figure 3.10: Plot of MSE for 1,000 Training and Test Data Splits

- The dashed line is $y = x$.
- The horizontal lines are mean of 1000 test MSE.
- The x-axis is the first 100 training MSE.
- The y-axis is the first 100 test MSE.

Figure 3.10 is the plot of 1,000 training and test mean squared errors for each of the models, with training error on x-axis and test error on the y-axis for each of the models. The dashed line is plotted at $y = x$ representing agreement between training and test error (MSE).
Figure 3.11: Mean of MSE for 1000 Training and Test Data Splits

- The x-axis is the mean of 1000 training MSE.
- The y-axis is the mean of 1000 test MSE.
- The dashed line is \( y = x \).

Figure 3.11 is the plot of the average of 1,000 training and test mean squared errors. The x-axis is the mean of training errors and the y-axis is the mean of test errors. Overall, random forest and bagging have the lowest training error but big discrepancies between training and test errors. Elastic net, lasso and tree with pruning have minimum discrepancies between training and test errors. The training and test errors for BRSS are relatively low compared to lasso, elastic net, and tree with pruning but with minor
discrepancy between training and test errors.

Figure 3.12: Training and Test R-squared

- The red circles are training r-squareds.
- The blue squares are test r-squareds.
- The x-axis is the number of training and test data splits.
- The y-axis is multiple r-squareds.
Figure 3.13: Training and Test R-squared

- The red circles are training r-squareds.
- The blue squares are test r-squareds.
- The x-axis is the number of training and test data splits.
- The y-axis is multiple r-squareds.
- Model brst uses adjusted r-squared.
Figures 3.12 and 3.13 are plots of training and test $R^2$ for models both FB, forward, backward, lasso, elastic net, decision tree, tree with pruning, bagging, and random forest across 1,000 splits of training and test data partitions. The BRSS model plot uses adjusted $R^2$ as specified before. The x-axis is the number of training and test splits. The red circles are training $R^2$ and the blue squares are the test $R^2$. Model summaries of bagging and random forest have the highest discrepancy between training and test $R^2$, therefore, these two models are consistently over optimistic in making predictions to test data. Stepwise procedures have similar results with the backward method having the highest training $R^2$ on average. The points for training and test $R^2$ for lasso, elastic net and BRSS are almost perfectly overlapping; thus these models have minimum discrepancy between training and test $R^2$ which is a highly desirable characteristic.

Figure 3.14 shows the relationship between training and test $R^2$ for each of the 1,000 training and test partitions. The x-axis is the training $R^2$ and the y-axis is test $R^2$. The dashed line is plotted at $y = x$. Points below the line represent models which have higher training $R^2$ relative to test $R^2$. Approximately half of the points for models BRSS, lasso, elastic net, and tree with pruning are above the the line. Thus, these models perform better on average because they systematically do not over- or under-estimate model fit whereas the stepwise procedures do. Models such as bagging and random forest have the highest training $R^2$ but relatively low test $R^2$ so these two models overfit the training data as well.

Figure 3.15 is the summary of all 1,000 partitions displaying the mean of 1,000 $R^2$ points for each of the methods. The x-axis is training $R^2$ and the y-axis is test $R^2$. The dashed line is plotted at $y = x$ as a reference line. This plot demonstrates the performance in overall out of sample prediction and fit of each of the models. The points for BRSS, pruning, lasso, and elastic net are falling at slightly different points on the $y = x$ line; so
these models have minimum overfitting and prediction error. Model BRSS is the best in this group because it also has the highest training and test $R^2$ on average. The elastic net ($\alpha = 0.05$) point overlapped with lasso so both models perform equally well on average.

Stepwise forward and bothFB models yielded higher training $R^2$ but these models did not do as well in predicting test data. Bagging and random forest models had the highest training $R^2$ but these models did not do as well in test data set.
Figure 3.15: The plot of the average of 1,000 training $R^2$ versus the mean of 1,000 test $R^2$ for each of the models: forward, backward, both simultaneously forward and backward, lasso, elastic net, decision tree, tree with pruning, bagging and random forest. Model best random subset selection is the mean of 1,000 training adjusted $R^2$ versus the mean of 1,000 test adjusted $R^2$.

- The dashed line is $y = x$.
- The x-axis is the mean of 1000 training r-squared.
- The y-axis is the mean of 1000 test r-squared.
- Model BRSS is the mean of 1000 training/test adjusted r-squared.
Chapter 4

Conclusion

We have shown that commonly used model selection methods can suffer from overfitting; the worst offenders were random forest and bagging. Stepwise procedures are not quite as bad but still suffer from overfitting. Methods that were consistently not overfitting the data included lasso, elastic net, tree with pruning, and BRSS (best random subset selection). Among this set of methods BRSS tended to give the best predictions on average. It had the highest test r-squared and was unbiased in its assessment of model fit to external data (see Figure 4.1). BRSS also tended to have fewer variables selected making it easier for drawing inference and producing more parsimonious models. We also saw some evidence that combining important variables selected across different methods (count 4, count 5) may be useful as well.

The socio-demographic variables most consistently selected by the models were: percentage of population that is African American, percentage of households with retirement income in 1989, percentage of people 16 and over who are employed in manufacturing jobs, percentage of people 16 and over who are employed in professional services, percentage of kids in family housing with 2 parents, number of kids born to never married
Figure 4.1: The plot of the average of 1,000 training $R^2$ versus the mean of 1,000 test $R^2$ for each of the models: forward, backward, both simultaneously forward and backward, lasso, elastic net, decision tree, tree with pruning, bagging, random forest. Model best random subset selection is the mean of 1,000 training adjusted $R^2$ versus the mean of 1,000 test adjusted $R^2$.

- The dashed line is $y = x$.
- The x-axis is the mean of 1000 training r-squared.
- The y-axis is the mean of 1000 test r-squared.
- Model BRSS is the mean of 1000 training/test adjusted r-squared.
parents, percentage of people in dense housing, and median owners cost as a percentage of household income (for owners without a mortgage). Similar to the other countries we investigated, we also found that measures of employment, urbanization/density, income, poverty, and family structure were all consistently associated with crime. Although these were the most consistently selected factors, it is important to not conclude causal relationships and acknowledge that there are likely other confounding variables as well. It is also important to note that our database was aggregated at the city level and our results are not meant to generalize to individuals.
Bibliography


Appendix: R Code Implementing Best Random Subset Selection

This appendix included the r-code for best random subset selection model. Here $M$ is the number of times to perform BRSS model selection. $p$ is the maximum number of predictors to use in BRSS models, and $N$ is the number of models fit on the training data set for each $p$.

# get data ready
# ------------Read in data and remove all missing-------------
setwd("C:/Users/Joyce/Desktop/Thesis")
mythesis <- read.csv("CrimeData.csv",stringsAsFactors=FALSE)
CompleteData <- na.omit(thes)
dim(mythesis)
dim(CompleteData)

CompleteData<-subset(CompleteData,select=-c(autoTheftPerPop, 
nonViolPerPop, 
burg1PerPop, 
robblPerPop, 
assaultPerPop, 
assaults, 
burglaries, 
larcenies, 
autoTheft, 
arsonsPerPop, 
larcPerPop, 
rapesPerPop, 
rapesPerPop,
# repleat brss 1000 times (M=1000)

# number of test/training splits:
M <- 1000

# store the results of M splits
brss.train.aR2 <- numeric(M)
brss.test.aR2 <- numeric(M)
brss.full.aR2 <- numeric(M)
MSE.train.brss <- numeric(M)
MSE.test.brss <- numeric(M)
lm.train.brss <- list()
lm.test.brss <- list()
  length(lm.train.brss) <- M
  length(lm.test.brss) <- M

# store model: length M
max.mod <- list()
length(best.mod) <- M
set.seed(0406)

# loop over test/training splits:
for (j in 1:M) {
    # break into training and test set
    train.obs <- sample(nrow(CompleteData), 0.75*nrow(CompleteData))
    XY.train <- as.data.frame(CompleteData[train.obs,])
    Y.train <- subset(XY.train, select=ViolentCrimesPerPop)
    X.train <- subset(XY.train, select = -ViolentCrimesPerPop)
    XY.test <- as.data.frame(CompleteData[-train.obs,])
    Y.test <- subset(XY.test, select=ViolentCrimesPerPop)
    X.test <- subset(XY.test, select = -ViolentCrimesPerPop)

    # fit full model
    brss.full.aR2[j] <- summary(lm(XY.train$ViolentCrimesPerPop ~ as.matrix(X.train)))$adj.r.squared

    N <- 1000
    p <- 20
    best.mod <- list()
    length(best.mod) <- p
    train.adr2 <- numeric(p)
    test.adr2 <- numeric(p)

    # loop over number of predictors included in model: (set p=20)
    for(p in 1:p) {
        current.best.adr2 <- 0
        current.best.varused <- NULL

        # loop over random subsets of q predictors:
for (i in 1:N) {
  varused <- sample(ncol(X.train), p)
  lm.train <- lm(XY.train$ViolentCrimesPerPop ~
                 as.matrix(X.train[, varused]))
  adr2 <- summary(lm.train)$adj.r.squared
  if (adr2 > current.best.adr2) {
    current.best.varused <- varused
    current.best.adr2 <- adr2
  }
} # end i loop for random subsets

# store the variables used in the best model for the given p:
best.mod[[p]] <- current.best.varused

# store the training adjusted R2 for the best model for the given p:
train.adr2[p] <- current.best.adr2

# store the test adjusted R2 for the model with the
# best training aR2 for given q:
lm.test <- lm(XY.test$ViolentCrimesPerPop ~
              as.matrix(X.test[, current.best.varused]))
test.adr2[p] <- summary(lm.test)$adj.r.squared
} # end p loop for number of predictors

# obtain training and test aR2
brss.train.aR2[j] <- max(train.adr2)
brss.test.aR2[j] <- test.adr2[which.max(train.adr2)]
max.mod[[j]] <- best.mod[[which.max(train.adr2)]]

# store best training lm() model
lm.train.brss[[j]] <- lm(XY.train$ViolentCrimesPerPop ~ .,  
                         data=X.train[,best.mod[[which.max(train.adr2)]]])

# Find training MSE
yhat.train.brss <- predict(lm.train.brss[[j]],  
                           newdata=XY.train, type = "response")
TSS.train <- sum((Y.train$ViolentCrimesPerPop-  
                  mean(Y.train$ViolentCrimesPerPop))^2)
RSS.train.brss <- sum((Y.train$ViolentCrimesPerPop-yhat.train.brss)^2)
MSE.train.brss[j] <- (1/605)*RSS.train.brss

# Find test MSE
yhat.test.brss <- predict(lm.train.brss[[j]],  
                           newdata=XY.test, type = "response")
TSS.test <- sum((Y.test$ViolentCrimesPerPop-  
                 mean(Y.test$ViolentCrimesPerPop))^2)
RSS.test.brss <- sum((Y.test$ViolentCrimesPerPop-yhat.test.brss)^2)
MSE.test.brss[j] <- (1/202)*RSS.test.brss

} # end j loop for test/training splits