COMPARISON OF DIFFERENT STATISTICAL PREDICTION METHODS FOR FETAL GROWTH ABNORMALITIES

by

Hongqun Zhang

Submitted in partial fulfillment of the
requirements for the degree of
Master of Science

at

Dalhousie University
Halifax, Nova Scotia
August 2014

This is to the memory of my father Zhongren Zhang.

I dedicate my thesis work and a special feeling of gratitude to my loving parents Zhongren Zhang and Ruixia Song for your continuing inspiration and encouragement in my life.

I also dedicate my thesis to my siblings Weiqun Zhang and Wenqun Zhang and thank for always being there to support me.

I dedicate this work and give special thanks to my husband Yimin Liu and my daughter Yuhan Liu who have always been great resources of my motivation and inspiration.

Table of Contents

List of Tables

List of Figures

Abstract

The decision tree method has become very popular because it can efficiently accommodate a large amount of data with a mixture of different types of variables, missing data and many irrelevant predictors. Trees also can be graphically presented and easily explained. However, the weaknesses of the decision tree model are high variance and lower predictive accuracy. These problems have been substantially improved by the tree ensemble-based methods: random forests and boosting trees. In this study, tree and tree ensemble-based methods, as well as logistic regression are reviewed and are applied to the Nova Scotia Atlee Perinatal Database to predict fetal growth abnormalities such as infants with birth weight small for gestational age or large for gestational age. It was found that predictive accuracy of the boosted tree model is better than both random forests and decision trees, but this model does not show much improvement over logistic regression.

List of Abbreviations and Symbols Used

Symbols and Abbr.	Description
Y_i	Response variable
$\mathbf{x}_i$	Random vector for predictor variables
η	Logit function of probability
LR^+	Positive likelihood ratio
LR^-	Negative likelihood ratio
T	A classification or regression tree
$\tilde{T}$	The set of terminal nodes of T
T_τ	A branch rooted at node τ
$\tilde{T}_\tau$	Terminal nodes of T_τ
$R(T)$	Regression cost
$R_\alpha(T)$	Cost- complexity function
α	Complexity parameter
cp	Dimensionless complexity parameter
$G(\mathbf{x})$	Classifier for predictor variable $\mathbf{x}$
$L(y, f(\mathbf{x}))$	Loss function
ω_i	Weights applied to observations
α_m	Weights applied to classifiers
err_m	Weighted classification error rates
NSAPD	the Nova Scotia Atlee Perinatal Database
SGA	Infants with birth weight small for gestational age
LGA	Infants with birth weight large for gestational age
ROC	Receiver operating curve
AUC	Area under the ROC curve
AIC	Akaike information criterion
CART	Classification and regression tree
OOB	Out-of-bag samples

Acknowledgements

I wish to especially thank my supervisors Dr. David Hamilton and Dr. Stefan Kuhle for their very professional and great guidance, detailed and useful comments, encouragement and patience throughout this work.

My special thanks also go to Dr. Hong Gu for her helpful discussion and suggestions. My appreciations must go to Dr. Bruce Smith, Dr. Edward Susko, Dr. Keith R. Thompson, Dr. Michael Dowd for their excellent instruction on the courses that I took during my study at Dalhousie university.

I am grateful to Balagopal Pillai for his technical support and Paula Flemming and other members of staff for their assistance in the department office at Dalhousie University. I also thank Janet Slaunwhite and Neil Colosimo and other members of staff for their assistance and technical support at the IWK Health Centre.

Chapter 1

Introduction

1.1 Fetal growth abnormalities

Abnormal fetal growth may have both short-term and long-term consequences in neonates. Infants with abnormal birth weight account for the majority of mortality and morbidity in neonates born at term (Das, 2014). Infant growth is commonly assessed by comparing the infant's birth weight to that of an age- and sex-specific reference population of infants. An infant with a birth weight below the 10^{th} percentile (small for gestational age, SGA) or above the 90^{th} percentile (large for gestational age, LGA) for gestational age and sex is commonly considered to be at a higher risk for perinatal mortality and morbidity. In 2008, approximately 43,000 infants in Canada were born SGA or LGA, accounting for 8.1% and 11.1% of livebirths, respectively (Public Health Agency of Canada, 2012). Infants born SGA are at a higher risk for developing hypoglycaemia, hyperbilirubinaemia, respiratory distress, polycythaemia, thrombopenia, and necrotizing enterocolitis than their appropriate for gestational age counterparts. Infants that are LGA are at a higher risk for prolonged and complicated labour due to physical size and subsequent birth injury, the need for assisted delivery or cesarean section, asphyxia, meconium aspiration, and other postnatal problems (Minior, 1998 and Longo, 2012). Fetal growth is influenced by maternal health and health behaviors (such as excess pre-pregnancy weight and gestational weight gain, gestational diabetes, and smoking during pregnancy), e.g. obese mothers or mothers with high pregnancy weight gain are more likely to have a LGA baby, while mothers who smoke during pregnancy or have severe gestational diabetes are at higher risk to deliver a SGA infant (Nohr 2008, De Vader SR, 2007, Siega-Riz, 2009 and Van, 2001). Therefore, predicting (abnormal) birth weight for gestational age of infants can help clinicians plan for high risk deliveries and can inform policy makers how changes in maternal health and health behaviors influence rates of fetal growth abnormalities.

1.2 Diagnostic testing

SGA and LGA can be treated as qualitative responses. There are many possible classification techniques, or classifiers, that one might use to predict a qualitative response. Logistic regression is the most commonly used method in health research. In this study, we will explore more computer-intensive methods such as classification trees, random forests, and boosting for neonatal outcomes based on the same sets of predictors.

Classifiers will be evaluated by the properties of diagnostic tests, including sensitivity, specificity, positive predictive value, negative predictive value, and likelihood ratio. The first four measures have long been used in clinical epidemiology, but they only summarize the characteristics of a population, and are of limited value for use in individual patients. The likelihood ratio is more clinically useful since it can be interpreted at the individual patient level and allows clinicians to estimate the probability of disease for any individual patient given the test results. A clinically meaningful test will provide a high positive likelihood ratio (LR^{+}) and a small negative likelihood ratio (LR^{-}).

All these measures are related to true positive (TP), false positive (FP) (type I errors), false negative (FN) (type II errors) and true negative (TN) rates in the confusion matrix (Table 1.1).

		True Class		
		Disease^{+}	**Disease^{-}**	**Total**
Predicted Class	**Test^{+}**	True Positive (TP)	False Positive (FP)	P*
	Test^{-}	False Negative (FN)	True Negative (TN)	N*
	Total	P	N	

Table 1.1: Confusion Matrix.

In our study, "Disease^{+}" in Table 1.1, refers to the women who actually deliver SGA (or LGA) babies, while "Disease^{-}" refers to the women who do not actually deliver SGA babies. "Test^{+}" and "Test^{-}" stands for the predicted (or test) results of those having SGA (or LGA), and not having SGA (or LGA) babies, respectively.

Sensitivity is the proportion of those with disease who test positive (true positive rate). Specificity is the proportion of those without disease who test negative (true negative rate). Therefore, these rates can be expressed as

$$sensitivity = \frac{TP}{P} = \frac{TP}{TP + FN};$$

$$specificity = \frac{TN}{N} = \frac{TN}{FP + TN}.$$

The positive predictive value expresses the proportion of those with positive test results who truly have disease. The negative predictive value expresses the proportion of those with negative test results who truly do not have disease. In other words, the positive predictive value is the probability that they truly have disease given that a patient tests positive, and the negative predictive value is the probability that they truly do not have disease given that a patient tests negative. Thus, these rates can be expressed as

$$positive\ predictive\ value = \frac{TP}{P*} = \frac{TP}{TP + FP};$$

$$negative\ predictive\ value = \frac{TN}{N*} = \frac{TN}{TN + FN}.$$

The positive likelihood ratio is the probability of a person who has the disease testing positive divided by the probability of a person who does not have the disease testing positive. The negative likelihood ratio is the probability of a person who has disease testing negative divided by the probability of a person who does not have the disease testing negative. These likelihood ratios can be expressed by

$$LR^{+} = \frac{TP\ rate}{FP\ rate} = \frac{sensitivity}{1 - specificity};$$

$$LR^{-} = \frac{FN\ rate}{TN\ rate} = \frac{1 - sensitivity}{specificity}.$$

The higher the positive likelihood ratio and the lower the negative likelihood ratio, the better the classifier.

The classification techniques we will be considering produce a probability that the case is SGA or LGA. The threshold is the value of the probability beyond which the case is classified as SGA or LGA. Varying the classifier threshold changes its true positive and false positive rate. Therefore, the above measures from a diagnostic test rely on a single threshold to classify a test result as SGA or LGA. The optimal threshold will be selected based on the receiver operating curve (ROC).

The ROC curve is considered a useful tool for comparing different classifiers, since it takes into account all possible thresholds. The true positive rate (sensitivity) is plotted versus the false positive rate (1-specificity) across a series of cutoff points of a diagnostic test, and shows the trade-off between sensitivity and specificity when the decision threshold changes. A high threshold corresponds to (0,0), while a low threshold corresponds to (1,1) in the space. Since the upper left corner represents 100% sensitivity and 100% specificity, the optimal threshold to discriminate the two classes of subjects is the value corresponding to the point on the ROC curve nearest to the upper left corner.

The overall performance of a classifier, summarized over all possible thresholds, is given by the area under the ROC curve (AUC). The AUC ranges from 0.5 to 1. The upper and lower bounds correspond to the ROC curve passing through the upper left corner (0, 1) and the 45 degree diagonal line (which corresponds to a random guess), respectively. Hence, the larger the AUC, the better the performance of the test. Hosmer and Lemeshow (2000) pointed out that a general rule for assessment of discrimination performance for a classifier by AUC is that if $AUC = 0.5$: no discrimination; if $0.7 \leq AUC < 0.8$: acceptable discrimination ; if $0.8 \leq AUC < 0.9$: excellent discrimination; if $AUC \geq 0.9$: outstanding discrimination.

1.3 The dataset

The dataset that we will use to build the prediction models and compare prediction accuracy of different models is the Nova Scotia Atlee Perinatal Database (NSAPD). The NSAPD contains records on approximately 50,000 infants born in Nova Scotia between 2004 and 2012. The dataset selected for the purpose of the current study provides data on birth weight, gestational age, infant sex, and potential predictors of SGA/LGA such as maternal pre-pregnancy BMI, smoking, sociodemographics, and maternal health conditions during pregnancy. Ninety percent of the cases are randomly assigned into training data and the remaining 10% of the cases are used as test data. We will build models based on the training data and validate the model using the test data.

The primary outcomes for our study will be SGA and LGA. Infants will be categorized as SGA ($< 10^{\text{th}}$ percentile for gestational age and sex) vs. not SGA, and LGA ($> 90^{\text{th}}$ percentile for gestational age and sex) vs. not LGA using the Canadian birth weight reference values by Kramer et al (2001). We define two binary response variables sga and lga. We code $sga = 1$ as SGA and $sga = 0$ as not SGA. Similarly, we code $lga = 1$ as LGA and $lga = 0$ as not LGA. The predictors that we use are listed on Table 1.2.

The structure of the thesis is as follows. The logistic regression model is introduced in Chapter 2. We will use this model to predict the SGA and LGA birth weight categories. Chapter 3 provides a theoretical overview of the classification and regression tree model. In Chapter 4, the random forest is described. Chapter 5 describes the gradient boosting method. The main results are summarized and discussed, and suggestions for future work are made in Chapter 6.

Variable	Levels	Variable Name	Definition
Maternal age	Range X-Y	*matage*	Maternal age in years
Gestational diabetes	1 Yes, 2 No	*gdm*	
Pre-existing diabetes	1 Yes, 2 No	*dm*	
Rural residence	1 Yes, 2 No	*rural*	Rough estimate based on maternal residence postal code. Rural = '0' in the 2nd position
Area-level SES quintiles	1 Q1, 2 Q2, 3 Q3, 4 Q4, 5 Q5	*ses*5	Area-level income quintile based on Census information
Birth weight for GA	1 AGA, 2 SGA, 3 LGA	*bwga*	Cutoffs are based on Canadian reference values for birth weight-for-gestational age and sex (Kramer et al. 2000).
Parity	0 0, 1 I, 2 II, 3 III+	*parity*	Number of previous deliveries
Previous C-section	0 No, 1 Yes	*prvcs*	
Previous birth < 2500g	0 No, 1 Yes	*prvlbw*	
Previous birth > 4080g	0 No, 1 Yes	*prvbig*	
Smoking status on admission	0 Non-smoker 1 < 0.5 pack/day 2 >= 0.5 pack/day 3 Unknown	*smk*	
Pre-pregnancy weight status	1 Underweight 2 Normal 3 Overweight 4 Obese	*ppwtstat*	Estimated from pre-pregnancy weight and average women's height
Preganancy weight gain	0 Adequate 1 Inadquate 2 Excessive	*pwtgain*3	Difference between pre-pregnancy and delivery weight, categorized as per Institute of Medicine recommendations (Rasmussen and Yaktine 2009).
Previous gestational diabetes	0 No, 1 Yes	*prvgdm*	
Pre-existing hypertension	0 No, 1 Yes	*hyp*	
Pregnancy-induced hypertension	0 No, 1 Yes	*pihyp*	
Chemical abuse	0 No, 1 Yes	*chabus*	
Psychiatric illness during pregnancy	0 No, 1 Yes	*psych*	

Table 1.2: Variables derived from the Nova Scotia Atlee Perinatal Database (NSAPD).

Chapter 2

Logistic Regression

Logistic regression is the most commonly used method for predicting a qualitative response in health research. We will introduce logistic regression theory first and then apply it to predict SGA and LGA.

2.1 Theory

For data $(y_i, \mathbf{x}_i), i = 1, \ldots, n$, $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})$, we assume that the response Y_i for each of observations has the Bernoulli distribution

$$p(Y_i = y_i) = p_i^{y_i}(1 - p_i)^{1-y_i}, \tag{2.1}$$

where the parameters

$$\boldsymbol{p} = (p_1, \ldots, p_n)'$$

must be estimated from the data. The logistic regression model is established by introducing the logistic function

$$p_i = \frac{exp(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij})}{1 + exp(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij})}, \tag{2.2}$$

to relate the probability to the predictors, $\mathbf{x}_i$, where

$$\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)'$$

is the $p + 1$ dimensional vector of parameters to be estimated.

We maximize the likelihood function to estimate $\boldsymbol{\beta}$. If $Y_1, Y_2, \ldots, Y_n$ are independent, the likelihood function is given by

$$L = \prod_{i=1}^{n} {p_i}^{y_i}(1 - p_i)^{1-y_i}. \tag{2.3}$$

The log likelihood function for the parameters $\boldsymbol{\beta}$ is given by

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{n} [y_i \eta_i - log(1 + e^{\eta_i})], \tag{2.4}$$

where η_i is the logit function of p_i defined by

$$\eta_i = logit(p_i) = log(\frac{p_i}{1 - p_i}).$$

In logistic regression the logit function of p_i is linearly related to the predictors,

$$\eta_i = \mathbf{x}_i' \boldsymbol{\beta}.$$

To maximize the log likelihood function $l(\boldsymbol{\beta})$, we set its derivative to zero and solve the $p + 1$ likelihood equations

$$\frac{\partial l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \sum_{i=1}^{n} [\mathbf{x}_i'(y_i - p_i(\boldsymbol{\beta}))] = \mathbf{0}. \tag{2.5}$$

When the first component of $\mathbf{x}_i'$ is 1, the first likelihood equation specifies that $\sum\limits_{i=1}^{n} y_i = \sum\limits_{i=1}^{n} p_i(\boldsymbol{\beta})$. This means that the sum of the fitted probabilities must equal the number of cases for $y_i = 1$, or that the average fitted probability must equal the proportion of cases for $y_i = 1$ in the dataset.

To solve the likelihood equations (2.5), we use the Newton-Raphson algorithm, which is equivalent to the iteratively reweighted least squares (IRLS) method. Thereby the maximum likelihood estimator of $\hat{\boldsymbol{\beta}}$ of $\boldsymbol{\beta}$ is obtained.

By substituting $\hat{\boldsymbol{\beta}}$ for $\boldsymbol{\beta}$ in equation (2.2), for subject i, we estimate a woman's risk of having an SGA baby according to

$$\hat{p}_i = \frac{exp(\hat{\beta}_0 + \sum_{j=1}^{p} \hat{\beta}_j x_{ij})}{1 + exp(\hat{\beta}_0 + \sum_{j=1}^{p} \hat{\beta}_j x_{ij})}, \tag{2.6}$$

For any threshold r $(0 \leq r \leq 1)$, we calculate the number of subjects for which $\hat{p}_i > r$ and assess the performance of the logistic regression model as a diagnostic test.

2.2 Results

2.2.1 SGA

We start with the univariate analysis for every predictor listed in Table 1.2. The univariate analysis is based on simple logistic regression and the χ^2 test using the training data. The test statistic, degrees of freedom (df), and the P value are listed in Table 2.1 in order of ascending P value and decreasing change in deviance.

Variable	df	Odds Ratio	95% CI	Change in Deviance	P value
smk	3			587.86	<2.2e-16
*pwtgain*3	2			486.27	<2.2e-16
prvbig	1	0.11	(0.077, 0.162)	294.34	<2.2e-16
ppwtstat	3			227.84	<2.2e-16
parity	3			122.15	<2.2e-16
prvlbw	1	2.23	(1.910, 2.580)	91.614	<2.2e-16
matage	1	0.97	(0.963, 0.975)	94.954	2.20e-16
prvcs	1	0.63	(0.554, 0.716)	55.275	1.05e-13
*ses*5	4			59.981	2.93e-12
pihyp	1	1.59	(1.384, 1.817)	40.107	2.40e-10
hyp	1	1.94	(1.400, 2.633)	14.542	0.0001371
chabus	1	2.58	(1.623, 3.921)	14.415	0.0001466
prvgdm	1	0.70	(0.460, 1.004)	3.7639	0.05237
gdm	1	0.81	(0.665, 0.985)	4.4507	0.03489
dm	1	0.70	(0.430, 1.065)	2.7332	0.09828
rural	1	1.05	(0.967, 1.131)	1.263	0.2611
psych	1	1.00	(0.728, 1.342)	6.78e-05	0.9934

Table 2.1: Univariate regression analysis for SGA.

Smoking status on admission (*smk*), pregnancy weight gain (*pwtgain*3), previous births with high birth weight (*prvbig*), pre-pregnancy weight status (*ppwtstat*), number of deliveries (*parity*) and previous births with low birth weight (*prvlbw*) showed the strongest association with having an SGA birth, with the lowest P values and largest change in deviance. Maternal age (*matage*), number of previous C-sections (*prvcs*), area-level SeS quintiles (*ses*5), pregnancy induced hypertension (*pihyp*), pre-existing hypertension (*hyp*), chemical abuse (*chabus*) and gestational diabetes (*gdm*) are less important than the previous variables with larger P values and smaller changes in deviance, but are still significant. Previous gestational diabetes (*prvgdm*), pre-existing diabetes (*dm*), rural residence (*rural*) and psychiatric illness during pregnancy (*psych*) are not significant at the 0.05 level.

The odds ratio with confidence interval for the 1 *df* predictors are added in the above table. The 95 % CIs of the odds ratio for *prvgdm*, *dm*, *rural* and *psych* include 1, meaning that these predictors are not significant at the 0.05 level.

Multiple logistic regression was then used on all candidate variables listed in Table 1.2 to determine how they jointly predict SGA. The variables are listed in ascending order of P values from the univariate analysis. The analysis of deviance table for this model is shown in Table 2.2.

Variables	*df*	Change in Deviance	P value
smk	3	587.86	< 2.2e-16
*pwtgain*3	2	421.01	< 2.2e-16
prvbig	1	257.20	< 2.2e-16
ppwtstat	3	94.86	< 2.2e-16
parity	3	139.23	< 2.2e-16
prvlbw	1	79.18	< 2.2e-16
matage	1	1.73	0.187908
prvcs	1	6.96	0.008344
*ses*5	4	18.28	0.001087
pihyp	1	82.33	< 2.2e-16
hyp	1	26.47	2.676e-07
chabus	1	1.66	0.197930
prvgdm	1	1.18	0.277964
gdm	1	2.69	0.100960
dm	1	2.20	0.137900
rural	1	1.19	0.274372
psych	1	1.14	0.286648

Table 2.2: Analysis of deviance table of the full regression model without interactions for SGA.

The first six covariates: *smk*, *pwtgain*3, *prvbig*, *ppwtstat*, *parity* and *prvlbw* are still the most important variables, due to their large change in deviance and low P values. These changes in deviance are incremental, and represent the effect of a predictor after the previous variables are included in the model. *prvcs*, *ses*5, *pihyp* and *hyp* are also statistically significant when they are added to the model. However, the variables *matage*, *chabus*, *prvgdm*, *gdm*, *dm*, *rural* and *psych* are not statistically significant when they are included in the model. Before excluding the insignificant terms from future consideration, interaction terms were added to the model and backward elimination was performed using the **step** function in R. The Akaike information criterion (AIC) was applied in the **step** function. It deals with the trade-off between the goodness of fit of the model and the complexity of the model. At each step, the candidate for dropping is the variable with the smallest AIC score. The final model is the one with the minimum AIC score.

All possible two-way interactions were created from four biologically plausible and significant variables: smoking status on admission (*smk*), pre-pregnancy weight status (*ppwtstat*), pregnancy weight gain (*pwtgain*3) and parity (*parity*). Each of these interaction terms was added separately to the model to determine its significance. Three of the interaction terms were statistically significant and retained in the model: one between *smk* and *pwtgain*3, one between *smk* and *parity* and one between *smk* and *pwtgain*3. These are then added to the model, and the analysis of deviance table is displayed in Table 2.3.

Variable	*df*	Change in Deviance	*P* value
prvbig	1	294.34	< 2.2e-16
prvlbw	1	83.44	< 2.2e-16
matage	1	77.19	< 2.2e-16
prvcs	1	24.35	8.045e-07
*ses*5	4	37.87	1.191e-07
pihyp	1	36.40	1.608e-09
hyp	1	18.46	1.732e-05
chabus	1	9.03	0.002656
prvgdm	1	1.07	0.301254
gdm	1	1.81	0.178358
dm	1	3.56	0.059104
rural	1	0.69	0.406151
psych	1	0.20	0.6528
smk	3	471.96	0.652803
*pwtgain*3	2	428.00	< 2.2e-16
parity	3	139.64	< 2.2e-16
ppwtstat	3	97.15	< 2.2e-16
smk:*pwtgain*3	6	9.02	0.172249
smk:*parity*	9	12.14	0.205642
*pwtgain*3:*ppwtstat*	6	9.91	0.128447

Table 2.3: Analysis of Deviance Table of the Logistic Regression Model with Interactions for SGA.

None of the three interaction terms are significant when they are added to the model sequentially. Furthermore an overall test for the three interactions is not statistically significant ($\Delta D = 31.07$, $df = 21$, $P = .07$).

Backward elimination for the variable selection using AIC was then carried out. The final model obtained is

$$\begin{aligned} log[\frac{p(sga=1)}{1-p(sga=1)}] =& \hat{\beta}_0 + \hat{\beta}_1\ prvbig + \hat{\beta}_2\ prvlbw + \hat{\beta}_3\ matage + \hat{\beta}_4\ prvcs + \hat{\beta}_5\ ses5 \\ &+ \hat{\beta}_6\ pihyp + \hat{\beta}_7\ hyp + \hat{\beta}_8\ gdm + \hat{\beta}_9\ dm + \hat{\beta}_{10}\ smk \\ &+ \hat{\beta}_{11}\ pwtgain3 + \hat{\beta}_{12}\ parity + \hat{\beta}_{13}\ ppwtstat \end{aligned} \tag{2.7}$$

where the parameters $\hat{\beta}_i, i = 0, 1, \ldots, 13$ are given in Table 2.4.

The analysis of deviance for the final model is demonstrated in Table 2.5. All but two of the terms are significant at the 0.05 level.

From Table 2.4, the variables most strongly associated with SGA are *smk*, *hyp*, *prvlbm* and *pihyp*. The odds ratio for *smk* $<$ 0.5 pack/day (OR 2.60, 95%CI: 2.340-2.878), *smk* $>=$ 0.5 pack/day (OR 2.90, 95%CI: 2.541-3.285) and *smk*Unknown (OR 1.94, 95%CI: 1.646-2.284) indicate that when fixing other variables in the model, women who smoke less than 0.5 pack/day have an odds of delivering an SGA infant that is 2.60 times that of women who do not smoke, and women who smoke 0.5 pack/day or more have an odds of delivering SGA babies that is 2.90 times that of women who do not smoke. In addition, women whose smoking status is unknown have higher odds (1.94) of delivering an SGA infant than those who do not smoke.

The odds ratios for *pihyp* (OR 2.07, 95%CI: 1.792-2.385) and *hyp* (OR 2.69, 95%CI: 1.906-3.700) show that the mothers who have pregnancy-induced hypertension or pre-existing hypertension have much higher odds to deliver an SGA infants than women with a normal blood pressure. The odds of delivering an SGA infant for the mothers with pregnancy-induced hypertension is estimated to be 2.07 times higher than those who do not have pregnancy-induced hypertension. The odds of delivering an SGA for the mothers with pre-existing hypertension is estimated to be 2.69 times larger than those who do not have pre-existing hypertension.

Variable	Est.	Odds	95% CI	Std.	z	Pr(>|z|)
		Ratio		error	value	
(Intercept)	-2.37	0.09	(0.075, 0.116)	0.111	-21.42	< 2e-16
*prvbig*Yes	-1.73	0.18	(0.120, 0.253)	0.190	-9.09	< 2e-16
*prvlbw*Yes	0.79	2.20	(1.865, 2.587)	0.083	9.45	< 2e-16
matage	0.01	1.01	(0.999, 1.013)	0.004	1.65	0.098
*prvcs*Yes	-0.17	0.84	(0.732, 0.967)	0.071	-2.41	0.016
*ses*5Q2	-0.22	0.81	(0.720, 0.903)	0.058	-3.73	1.93e-4
*ses*5Q3	-0.13	0.88	(0.782 ,0.981)	0.058	-2.29	0.022
*ses*5Q4	-0.19	0.83	(0.742, 0.927)	0.057	-3.29	0.001
*ses*5Q5	-0.19	0.83	(0.738, 0.935)	0.061	-3.06	0.002
*pihyp*Yes	0.73	2.07	(1.792, 2.385)	0.073	9.98	< 2e-16
*hyp*Yes	0.99	2.69	(1.906, 3.700)	0.169	5.85	4.87e-09
*gdm*Yes	-0.18	0.83	(0.676, 1.019)	0.105	-1.74	0.082
*dm*Yes	-0.35	0.71	(0.431, 1.098)	0.238	-1.45	0.146
smk< 0.5 pack/day	0.95	2.60	(2.340, 2.878)	0.053	18.08	< 2e-16
smk>= 0.5 pack/day	1.06	2.90	(2.541, 3.285)	0.066	16.21	< 2e-16
*smk*Unknown	0.66	1.94	(1.646, 2.284)	0.083	7.97	1.61e-15
*pwtgain*3Inadquate	0.47	1.61	(1.454, 1.777)	0.051	9.27	< 2e-16
*pwtgain*3Excessive	-0.51	0.60	(0.552, 0.659)	0.046	-11.12	< 2e-16
*parity*I	-0.47	0.62	(0.567, 0.683)	0.048	-9.93	< 2e-16
*parity*II	-0.48	0.62	(0.548, 0.695)	0.061	-7.95	1.87e-15
*parity*III+	-0.56	0.57	(0.496, 0.650)	0.069	-8.2	2.47e-16
*ppwtstat*Underweight	0.49	1.62	(1.425, 1.849)	0.066	-7.31	2.72e-13
*ppwtstat*Overweight	-0.07	0.93	(0.844, 1.027)	0.05	-1.42	0.156
*ppwtstat*Obese	-0.29	0.75	(0.674, 0.831)	0.053	-5.42	< 5.88e-08

Table 2.4: Coefficients and odds ratios of the final regression model for SGA. The second column shows the estimates of coefficients.

Variable	*df*	Deviance	*P* value
prvbig	1	294.34	0.0000
prvlbw	1	83.44	0.0000
matage	1	77.19	0.0000
prvcs	1	24.35	0.0000
*ses*5	4	37.87	0.0000
pihyp	1	36.40	0.0000
hyp	1	18.46	0.0000
gdm	1	2.30	0.1295
dm	1	3.74	0.0532
smk	3	476.98	0.0000
*pwtgain*3	2	427.12	0.0000
parity	3	141.17	0.0000
ppwtstat	3	97.39	0.0000

Table 2.5: Analysis of deviance table of the final regression model for SGA.

The odds ratio for *prvlbw*Yes (OR 2.20, 95%CI: 1.865-2.587) shows that mothers who previously had an infant with a birth weight less than 2500g have 2.20 times the odds of delivering an SGA infant than those who did not have an infant with a birth weight less than 2500g.

Variables *pwtgain*3 and *ppwtstat* are also quite strongly associated with SGA birth. The odds ratios for *pwtgain*3Inadequate (OR 1.61, 95%CI: 1.425-1.849) and *pwtgain*3Excessive (OR 0.60, 95%CI: 0.552-0.659) indicate that mothers who have inadequate pregnancy weight gain have higher odds of delivering SGA babies than those who do have adequate weight gain, while mothers who have excessive pregnancy weight gain have lower odds of delivering SGA babies than those who have adequate pregnancy weight gain. The odds ratios for *ppwtstat*Underweight (OR 1.62, 95%CI: 1.425-1.849), *ppwtstat*Overweight (OR 0.93, 95%: 0.844-1.027) and *ppwtstat*Obese (OR 0.75, 95%CI: 0.674-0.831) reveal that women who are underweight before pregnancy have increased odds of delivering an SGA infant than those who have normal weight, while women who are overweight or obese before pregnancy have lower odds of delivering an SGA infant than those whose weight are normal .

matage was not statistically significantly associated with SGA birth in the final model.

*prvbig*Yes is strongly negatively associated with of SGA births. The odds ratio for *prvbig*Yes(OR 0.18, 95%CI: 0.120-0.253) shows that mothers who have previously given birth to an infant with a birth weight greater than 4080 g have much lower odds of an SGA birth than those who have not had a infant larger than 4080 g birth weight previously. *parity* is also quite strongly negatively associated with SGA birth. The odds ratio for *parity* shows that multiparous women (i.e. with one or more previous pregnancies) have lower odds to deliver an SGA infant than nulliparous women (i.e. women in their first pregnancy).

Other variables such as *prvcs*Yes, *ses*5, *gdm*Yes, *dm*Yes are associated with a lower odds for women to have SGA babies. The odds ratio for *prvcs*Yes (OR 0.84, 95%CI: 0.732-0.967) indicates that mothers who previously had a C-section have lower odds of delivering an SGA infant than those who did not have previous C-sections. For variable *ses*5 (area-level income quintile based on the woman's residence census dissemination area), women in the 2nd to 5th quintile have lower odds of having an SGA baby than those in the lowest quintile. The odds ratio for *gdm*Yes (OR 0.83, 95%CI: 0.675-1.019) indicates that the odds ratio of delivering SGA for mothers who have gestational diabetes is lower than that for mothers who do not have gestational diabetes. The odds ratio for *dm*Yes (OR 0.71, 95%CI: 0.431-1.098) indicates that the mothers who have pre-existing diabetes have lower odds of delivering an SGA infant than those who do not have pre-existing diabetes.

In order to evaluate overall performance of the final logistic regression model for SGA and to identify the optimal threshold for the predicted probability, the ROC curve of this model on the test data is plotted in Figure 2.1.

Figure 2.1 illustrates the true positive rate (sensitivity) as a function of false positive rate (1-specificity) when the threshold for predicted probability changes. Because the curve is above the line of equality, this ROC curve demonstrates that the logistic regression model improves the prediction precision over a random prediction. The area under the ROC curve (AUC) is 0.70, indicating that the model is acceptable based on the general rule for AUC introduced in Section 1.2 (Hosmer and Lemeshow, 2000).

The optimal threshold for predicted probability is calculated by minimizing the distance between the left upper corner (the (0,1) point) and the point on the ROC

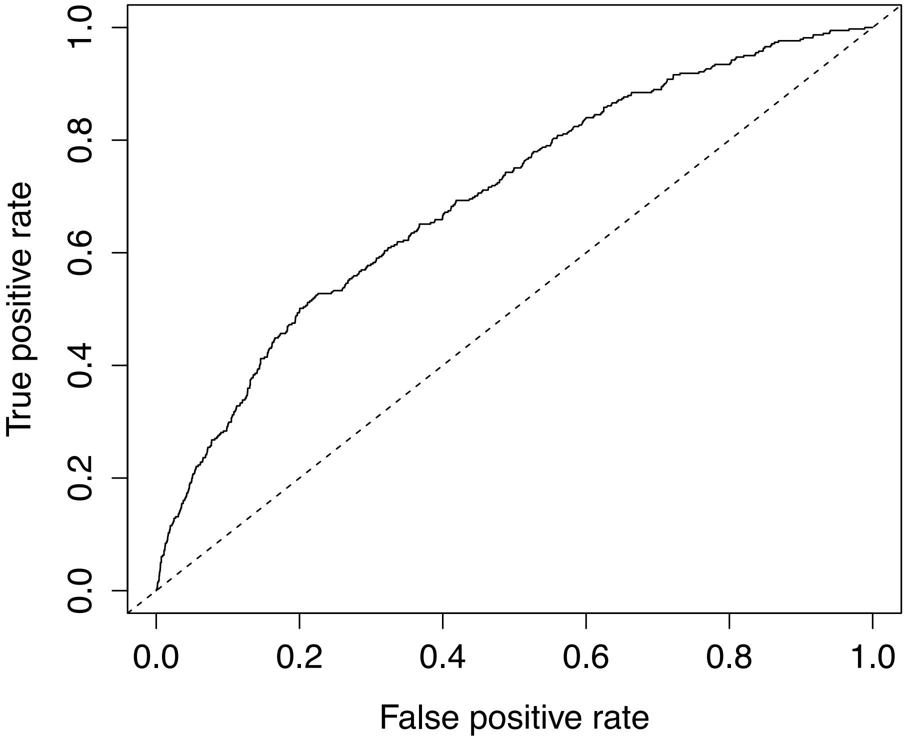

Figure 2.1: ROC curve for SGA.

curve. The new threshold we obtained is 0.065. In other words, we assign an observation to class SGA if $p_i(sga = 1) > 0.065$.

The characteristics of the diagnostic test that result from taking this approach are shown in Table 2.6.

	Disease+	Disease-	Total
Test+	248	1657	1905
Test-	133	2854	2987
Total	381	4511	4892

Point estimates and 95% CIs		
Apparent prevalence	0.39	(0.38, 0.40)
True prevalence	0.08	(0.07, 0.09)
Sensitivity	0.65	(0.60, 0.70)
Specificity	0.63	(0.62, 0.65)
Positive predictive value	0.13	(0.12, 0.15)
Negative predictive value	0.96	(0.95, 0.96)
Positive likelihood ratio	1.77	(1.63, 1.93)
Negative likelihood ratio	0.55	(0.48, 0.63)

Table 2.6: The confusion matrix and diagnostic summaries for SGA with threshold 0.065.

With the optimal threshold, the logistic regression model predicts that 1905 individuals will deliver SGA babies. Of the 381 individuals who actually deliver SGA, the logistic regression model correctly predicts 248, or sensitivity is 65%. However, 1657 individuals who are not in the SGA class are incorrectly classified. As a result, the overall error rate is 0.3659.

If the threshold of predicted probability changes to 0.051, the sensitivity increases from 0.65 to 0.80. However, this improvement comes at cost: the specificity decreases from 0.63 to 0.45. This demonstrates the trade-off between sensitivity and specificity of the diagnostic test with the threshold probability change.

The positive likelihood ratio (1.77) and the negative likelihood ratio (0.55) demonstrate that a mother with an SGA infant is about 1.77 times as likely to have a positive test than a mother without an SGA infant, and the probability of having a negative test for a mother with an SGA infant is 0.55 or about one half of that of a mother without an SGA infant.

2.2.2 LGA

In this section, we analyze the LGA outcome using logistic regression models. We start with the univariate analysis for every predictor listed in Table 1.2. The univariate analysis is based on the simple logistic regression and the chi-square test using the training data. The degrees of freedom, the odds ratio with 95% CI, the test statistic and the P value are listed in Table 2.7 in order of ascending P values and decreasing change in deviance.

Variable	*df*	Odds Ratio	95% CI	Change in Deviance	P value
prvbig	1	5.27	(4.886, 5.692)	1630.5	<2.2e-16
ppwtstat	3			853.31	<2.2e-16
*pwtgain*3	2			814.06	<2.2e-16
smk	3			471.56	<2.2e-16
parity	3			256.46	<2.2e-16
matage	1	1.03	(1.029, 1.039)	201.1	<2.2e-16
gdm	1	2.15	(1.934, 2.394)	177.04	<2.2e-16
prvcs	1	1.62	(1.508, 1.737)	167.75	<2.2e-16
dm	1	3.97	(3.236, 4.858)	153.87	<2.2e-16
prvlbw	1	0.51	(0.425, 0.615)	59.793	1.05e-14
prvgdm	1	2.10	(1.717, 2.531)	49.197	2.32e-12
rural	1	0.93	(0.874, 0.981)	6.929	0.008481
chabus	1	0.55	(0.298, 0.942)	4.8401	0.0278
psych	1	0.79	(0.614, 0.993)	4.0725	0.04359
hyp	1	1.33	(1.006, 1.726)	4.0133	0.04514
pihyp	1	1.07	(0.951, 1.193)	1.2233	0.2687
*ses*5	4			4.9813	0.2892

Table 2.7: Univariate regression analysis for LGA.

Previous births with high birth weight (*prvbig*), pre-pregnancy weight status (*ppwtstat*), pregnancy weight gain (*pwtgain*3), smoking status on admission (*smk*), number of deliveries (*parity*), maternal age (*matage*), gestational diabetes (*gdm*), number of previous C-sections (*prvcs*) and pre-existing diabetes (*dm*) showed the strongest association with having an LGA birth, with lowest P values and largest change in deviance. Previous births with low birth weight (*prvlbw*), previous gestational diabetes (*prvgdm*), rural residence (*rural*), chemical abuse (*chabus*), psychiatric illness during pregnancy (*psych*), and pre-existing hypertension (*hyp*) are less important than the previous variables with larger P values and smaller changes in deviance, but are still significant. Pregnancy induced hypertension (*pihyp*), area-level SeS quintiles (*ses*5) are not significant at the 0.05 level.

The 95 % CIs of the odds ratio for *pihyp* includes 1, showing that it is not significant at the 0.05 level.

Multiple logistic regression was then used on all predictors listed in Table 1.2 to determine how they jointly predict LGA. The variables are in ascending P values from the univariate analysis. The analysis of deviance table for this model is shown in Table 2.8.

	df	Change in Deviance	P value
prvbig	1	1630.47	0.0000
ppwtstat	3	623.66	0.0000
*pwtgain*3	2	613.84	0.0000
smk	3	367.73	0.0000
parity	3	72.18	0.0000
matage	1	9.05	0.0026
gdm	1	75.95	0.0000
prvcs	1	0.12	0.7292
dm	1	96.60	0.0000
prvlbw	1	41.90	0.0000
prvgdm	1	0.49	0.4833
rural	1	11.61	0.0007
chabus	1	0.04	0.8401
psych	1	4.27	0.0388
hyp	1	1.74	0.1870
pihyp	1	10.49	0.0012
*ses*5	4	2.02	0.7319

Table 2.8: Analysis of deviance table of the full regression model without interactions for LGA.

The first six covariates: *prvbig*, *ppwtstat*, *pwtgain*3, *smk*, *parity*, *gdm*, *dm* are still the most important variables, because of their large change in deviance and low P values. These changes in deviance are increamental, and represent the effect of a predictor after the previous variables are included in the model. *matage* becomes less important when combined with the previous variables. *prvlbw*, *rural*, *psych* and *pihyp* are also significant when they are added in the model. However, the variables *prvcs*, *prvgdm*, *chabus*, *hyp* and *ses*5 are not statistically significant at level of 0.05 when they are included in the model. Before excluding the non-significant terms from future consideration, interaction terms were added in the model and backward elimination was performed using AIC score. The final model is determined with the minimum AIC value.

All possible two-way interactions were created from 5 biologically plausible and significant variables: previous births with high birth weight (*prvbig*), pre-pregnancy weight status (*ppwtstat*), pregnancy weight gain (*pwtgain*3), smoking status on admission (*smk*) and number of deliveries (*parity*). Each of these interaction terms was added separately to the model to determine its significance. As a result, four of the interaction terms were statistically significant and retained in the model: one between *prvbig* and *pwtgain*3, one between *prvbig* and *ppwtstat*, one between *ppwtstat* and *pwtgain*3, and one between *smk* and *ppwtstat* are obtained. These are then added to the model, and the analysis of deviance table is displayed in Table 2.9.

Variable	*df*	Change in Deviance	P value
parity	3	256.46	0.0000
matage	1	97.42	0.0000
gdm	1	141.86	0.0000
prvcs	1	54.04	0.0000
dm	1	126.73	0.0000
prvlbw	1	120.42	0.0000
prvgdm	1	1.05	0.3050
rural	1	4.61	0.0318
chabus	1	2.23	0.1354
psych	1	5.76	0.0164
hyp	1	0.06	0.8128
pihyp	1	0.72	0.3958
*ses*5	4	2.30	0.6801
prvbig	1	1242.37	0.0000
*pwtgain*3	2	796.59	0.0000
ppwtstat	3	369.26	0.0000
smk	3	340.27	0.0000
prvbig:*pwtgain*3	2	7.88	0.0195
prvbig:*ppwtstat*	3	16.19	0.0010
*pwtgain*3:*ppwtstat*	6	40.83	0.0000
ppwtstat:*smk*	9	11.18	0.2636

Table 2.9: Analysis of deviance table of the full regression model with interactions for LGA.

Only one interaction term between *ppwtstat* and *smk* is not significant when it is included in the model. The other three interactions are highly significant when they are added in the model sequentially.

Backward elimination for variable selection using AIC was then carried out. The final model obtained is

$$\begin{aligned}
log[\frac{p(lga=1)}{1-p(lga=1)}] =& \hat{\beta}_0 + \hat{\beta}_1\ parity + \hat{\beta}_2\ matage + \hat{\beta}_3\ gdm + \hat{\beta}_4\ dm + \hat{\beta}_5\ prvlbw \\
&+ \hat{\beta}_6\ rural + \hat{\beta}_7\ psych + \hat{\beta}_8\ hyp + \hat{\beta}_9\ pihyp + \hat{\beta}_{10}\ prvbig \\
&+ \hat{\beta}_{11}\ pwtgain3 + \hat{\beta}_{12}\ ppwtstat + \hat{\beta}_{13}\ smk \\
&+ \hat{\beta}_{14}\ prvbig : ppwtstat + \hat{\beta}_{13}\ pwtgain3 : ppwtstat
\end{aligned} \tag{2.8}$$

where the parameters $\hat{\beta}_i$, $i = 0, 1, \ldots, 13$ are given in the Table 2.10

The analysis of deviance for the final model is demonstrated in Table 2.11. All but two of the terms are significant at the 0.05 level.

From Table 2.10, the most notable variables to influence LGA are *ppwtstat*, *prvbig*, *pwtgain*3 and their interactions. *dm* is also strongly associated with LGA births. The odds ratio for *dm*Yes (OR 3.262, 95%CI: 2.618-4.055) indicates that when fixing other variables in the model, the women who had pre-existing diabetes are estimated to have a odds that is about 3.262 times that of women of who did not have pre-existing diabetes.

The odds ratio for *prvbig*Yes in conjunction with *ppwtstat*Underweight is the product of odds ratios for *prvbig*Yes, *ppwtstat*Underweight and *prvbig*Yes:*ppwtstat*Underweight, which is $5.22 * 0.50 * 0.83 = 2.17$. This indicates that underweight mothers with a previous big baby have 2.17 times the odds of normal weight mothers with no previous big baby for having an LGA birth. Similarly, the odds ratio for *prvbig*Yes in conjunction with *ppwtstat*Overweight is $5.22 * 1.38 * 0.66 = 4.75$, indicating that overweight mothers with a previous big baby have even a higher odds, 4.75 times the odds of normal weight mothers with no previous big baby for having an LGA birth. The odds ratio for *prvbig*Yes in conjunction with *ppwtstat*Obese is $5.22 * 2.30 * 0.72 = 8.64$ shows that the obese mothers with a previous big baby have the highest odds, which is 8.64 times the odds of normal weight mothers with no previous big baby for having

Variable	Estimates	Odds Ratio	95% CI	Std. error	z value	Pr(>|z|)
(Intercept)	-2.72	0.07	(0.055, 0.079)	0.092	-29.48	< 2e-16
*parity*I	0.27	1.31	(1.219, 1.399)	0.035	7.60	3.05e-14
*parity*II	0.25	1.28	(1.174, 1.391)	0.043	5.67	1.43e-08
*parity*III+	0.24	1.27	(1.152, 1.397)	0.049	4.85	1.24e-06
matage	0.01	1.01	(1.000, 1.011)	0.003	2.02	0.043
*gdm*Yes	0.52	1.67	(1.488, 1.881)	0.060	8.61	< 2e-16
*dm*Yes	1.18	3.26	(2.619, 4.055)	0.112	10.60	< 2e-16
*prvlbw*Yes	-0.57	0.56	(0.463, 0.680)	0.098	-5.87	4.46e-09
*rural*Yes	-0.11	0.90	(0.846, 0.955)	0.031	-3.44	0.001
*psych*Yes	-0.25	0.78	(0.604, 0.999)	0.128	-1.92	0.055
*hyp*Yes	-0.21	0.81	(0.602, 1.069)	0.146	-1.46	0.144
*pihyp*Yes	-0.19	0.82	(0.730, 0.928)	0.061	-3.16	0.002
*prvbig*Yes	1.65	5.22	(4.541, 6.006)	0.071	23.18	2e-16
*pwtgain*3Inadquate	-0.57	0.56	(0.471, 0.672)	0.091	-6.315	2.69e-10
*pwtgain*3Excessive	0.69	1.99	(1.800, 2.209)	0.052	13.18	< 2e-16
*ppwtstat*Underweight	-0.68	0.50	(0.345, 0.713)	0.185	-3.70	2.13e-04
*ppwtstat*Overweight	0.32	1.38	(1.161, 1.627)	0.086	3.71	2.06e-04
*ppwtstat*Obese	0.83	2.30	(1.973, 2.676)	0.078	10.71	<2e-16
smk< 0.5 pack/day	-0.76	0.47	(0.413, 0.522)	0.060	-12.81	< 2e-16
smk>= 0.5 pack/day	-1.03	0.36	(0.301, 0.424)	0.087	-11.74	< 2e-16
*smk*Unknown	-0.36	0.70	(0.597, 0.813)	0.079	-4.56	5.03e-06
*prvbig*Yes:*ppwtstat*Underweight	-0.19	0.83	(0.413, 1.580)	0.340	-0.56	0.579
*prvbig*Yes:*ppwtstat*Overweight	-0.41	0.66	(0.542, 0.815)	0.104	-3.93	8.63e-05
*prvbig*Yes:*ppwtstat*Obese	-0.33	0.72	(0.593, 0.864)	0.096	-3.50	4.73e-4
*pwtgain*3Inadquate:*ppwtstat*Underweight	0.59	1.81	(0.973, 3.278)	0.308	1.92	0.054
*pwtgain*3Excessive:*ppwtstat*Underweight	0.46	1.59	(1.056, 2.450)	0.214	2.17	0.030
*pwtgain*3Inadquate:*ppwtstat*Overweight	0.02	1.02	(0.709, 1.452)	0.183	0.11	0.913
*pwtgain*3Excessive:*ppwtstat*Overweight	-0.09	0.92	(0.763, 1.102)	0.094	-0.93	0.350
*pwtgain*3Inadquate:*ppwtstat*Obese	0.29	1.35	(1.045, 1.735)	0.129	2.29	0.022
*pwtgain*3Excessive:*ppwtstat*Obese	-0.29	0.75	(0.634, 0.887)	0.086	-3.37	7.58e-4

Table 2.10: Coefficients and odds ratio of the final regression model for LGA.

Variable	df	Change in Deviance	P value
parity	3	256.46	0.0000
matage	1	97.42	0.0000
gdm	1	141.86	0.0000
dm	1	129.72	0.0000
prvlbw	1	109.57	0.0000
rural	1	4.55	0.0329
psych	1	5.61	0.0178
hyp	1	0.07	0.7956
pihyp	1	0.49	0.4844
prvbig	1	1299.43	0.0000
*pwtgain*3	2	798.26	0.0000
ppwtstat	3	372.67	0.0000
smk	3	342.97	0.0000
prvbig:*ppwtstat*	3	17.65	0.0005
*pwtgain*3:*ppwtstat*	6	43.33	0.0000

Table 2.11: Analysis of deviance table of the final regression model for LGA.

an LGA birth.

The odds ratio for *pwtgain*3Inadequate in conjunction with *ppwtstat*Underweight is $0.56 * 0.50 * 1.81 = 0.51$, which shows that underweight mothers who have inadequate pregnancy weight gain have 0.51 times the odds of delivering an LGA infant than those normal weight mothers who have adequate pregnancy weight gain. The odds ratio for *pwtgain*3Excessive in conjunction with *ppwtstat*Underweight is $1.99 * 0.50 * 1.59 = 1.58$, which shows that underweight mothers who have excessive pregnancy weight gain have a 1.58 times the odds, of delivering an LGA infant than those normal weight mothers who have adequate pregnancy weight gain. The odds ratio for *pwtgain*3Inadequate in conjunction with *ppwtstat*Overweight is $0.56 * 1.38 * 1.02 = 0.79$, which shows that overweight mothers who have inadequate pregnancy weight gain have 0.79 times the odds of delivering an LGA infant than those normal weight mothers who have adequate pregnancy weight gain. The odds ratio for *pwtgain*3Excessive in conjunction with *ppwtstat*Overweight is $1.99*1.38*0.92 = 2.53$, which shows that overweight mothers who have excessive pregnancy weight gain have 2.53 times the odds of delivering an LGA infant than those normal weight mothers who have adequate pregnancy weight gain. The odds ratio for *pwtgain*3Inadequate in conjunction with *ppwtstat*Obese is $0.56 * 2.30 * 1.35 = 1.74$, indicating that obese mothers with inadequate pregnancy weight gain have the odds of delivering an LGA infant 1.74 times that of those normal weight mothers with adequate pregnancy weight gain. The odds ratio for *pwtgain*3Excessive in conjunction with *ppwtstat*Overweight is $1.99 * 2.30 * 0.75 = 3.43$, which reveals that obese mothers with excessive pregnancy weight gain have the highest odds, 3.43 times the odds, of delivering an LGA infant than those normal weight mothers with adequate pregnancy weight gain.

parity and *gdm* are also associated with LGA births. The odds ratio for *parity* shows that multiparous women (i.e. with one or more previous pregnancies) have higher odds to deliver an LGA infant than nulliparous women (i.e. women in their first pregnancy). The odds ratio for *gdm*Yes (OR 1.67, 95%CI: 1.488-1.881) indicates that the odds ratio of delivering LGA babies for mothers who have gestational diabetes is higher than that for mothers who do not have gestational diabetes.

matage was statistically significantly associated with LGA birth in the final model. The odds ratio for *matage* (OR 1.01, 95%CI: 1.000-1.011) indicates that when maternal age increases by 1 year, the odds of having LGA babies has a very small increase.

smk and *prvlbw* are strongly negatively associated with LGA births. The odds ratio for *smk* < 0.5 pack/day (OR 0.47, 95%CI: 0.413-0.522), *smk* $>= 0.5$ pack/day (OR 0.36, 95%CI: 0.301-0.424) and *smk*Unknown (OR 0.70, 95%CI: 0.597-0.813) indicate that women who smoke less than 0.5 pack/day have lower odds of delivering an LGA infant than those who do not smoke, and women who smoke 0.5 pack/day or more have lower odds of delivering LGA babies than those who do not smoke. In addition, women whose smoking status is unknown have lower odds of delivering an LGA infant than those who do not smoke. The odds ratio for *prvlbw*Yes (OR 0.56, 95%CI: 0.463-0.680) shows that mothers who previously had an infant with a birth weight less than 2500g have lower odds of delivering an LGA infant than those who did not have an infant with a birth weight less than 2500g.

Other variables such as *rural*, *psych*, *hyp*, and *pihyp* are associated with lower odds for LGA births. The odds ratio for *rural*Yes (OR 0.90, 95%CI: 0.846-0.955) indicates that women who live in a rural area have lower odds than those who live in urban area. The odds ratio for *psych*Yes (OR 0.78, 95%CI: 0.604-0.999) indicates that women who have psychiatric illness during pregnancy have lower odds of delivering LGA babies than women who do not have psychiatric illness during pregnancy. The

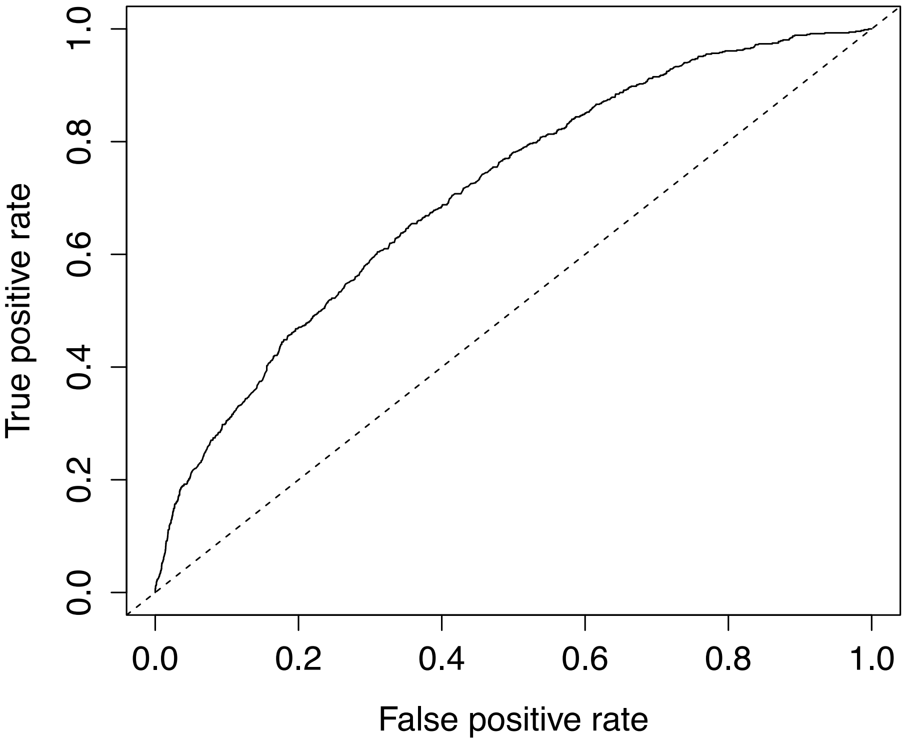

Figure 2.2: ROC Curve for LGA with interaction.

odds ratios for *hyp*Yes (OR 0.81, 95%CI: 0.602-1.069) and *pihyp*Yes (OR 0.82, 95%CI: 0.730-0.928) show that the mothers who have pre-existing hypertension or pregnancy-induced hypertension have lower odds to deliver an LGA infant than women with a normal blood pressure .

The ROC curve of the final logistic regression model for LGA on the test data is plotted in Figure 2.2.

Figure 2.2 illustrates the true positive rate (sensitivity) as a function of false positive rate (1-specificity) when threshold changes. It demonstrates that the logistic regression model improves the prediction precision over a random prediction model. The area under the ROC curve is 0.7085, indicating that the model is acceptable. The new threshold we obtained is 0.153. In other words, instead of assigning an observation to the LGA class if $p_i(lga = 1) > 0.5$ holds, we could instead assign an observation to this class if $p_i(lga = 1) > 0.153$.

The results from taking this approach are shown in Table 2.12.

	Disease+	Disease-	Total
Test+	453	1416	1869
Test-	265	2758	3023
Total	718	4174	4892

Point estimates and 95% CIs		
Apparent prevalence	0.38	(0.37, 0.40)
True prevalence	0.15	(0.14, 0.16)
Sensitivity	0.63	(0.59, 0.67)
Specificity	0.66	(0.65, 0.68)
Positive predictive value	0.24	(0.22, 0.26)
Negative predictive value	0.91	(0.90, 0.92)
Positive likelihood ratio	1.86	(1.73, 1.99)
Negative likelihood ration	0.56	(0.51, 0.62)

Table 2.12: The confusion matrix and diagnostic summaries for LGA with threshold 0.153.

With the optimal threshold, the logistic regression model predicts that 1869 individuals will deliver LGA babies. Of the 718 individuals who deliver LGA babies, logistic regression model correctly predicts 453, or 63%. 1416 individuals who are not LGA are incorrectly classified. As a result, the overall error rate has increased to 0.3436.

The positive likelihood ratio (1.86) and the negative likelihood ratio decreases (0.56) indicate that a woman with an LGA birth is about 1.86 times as likely to have a positive test than a women without a LGA birth, and the probability of having a negative test for women with a LGA birth is 0.56 or about one half of that of those without a LGA birth.

Chapter 3

Classification and Regression Trees

The classification and regression tree model (CART) was introduced to statistics by Breiman, Friedman, Olshen, and Stone (1984). It is very popular because of its easy interpretability and intuitive graphical presentation. It can be used for solving both regression and classification problems. In CART, the predictor space is partitioned by recursive binary splitting according to some specific criteria into different cases, which is graphically represented by a tree. The predicted value of the response variable is calculated based on a simple model (often a constant) in the region where the observation falls.

In this chapter, we will introduce the process for building regression and classification trees and for achieving the optimal size tree by a pruning method. We define *sga* and *lga* as response variables, with 1 corresponding to SGA or LGA and 0 corresponding to not SGA or not LGA. Then we will use classification trees to predict SGA and LGA from the predictors.

3.1 Regression Trees

Following Breiman et al. (1984) and Hastie et al. (2009), we begin with building a regression tree and then find the optimal size of the tree.

For data $(y_i, \mathbf{x}_i)$, $i = 1, \ldots, n$ with p predictors $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})$, suppose we partition the predictor space into M regions $R_1, R_2, \ldots, R_M$. We have many ways to partition the predictor space to get different shapes for each region. However, in order to express all partitions by a tree with M terminal nodes, we partition the predictor space into high-dimensional rectangles or boxes by using recursive binary partitioning. At each step, we partition the region of the predictors into two regions where the division is parallel to one of the axes. The sub-partition is operated within existing partitions and is not allowed to cross them. The process of recursive binary splitting is elaborated as follows.

We first select the predictor x_j and splitting value s from all predictors and points along the region of each predictor and divide the predictor space into the pair of regions $R_1(j,s) = \{X|x_j \geq s\}$ and $R_2(j,s) = \{X|x_j < s\}$.

If we use the residual sum of squares (RSS) as our fitting criterion for a continuous response, the best estimator of the response variable y_i is the average over all observations in the same region as i.

Therefore, the values of j and s can be found by solving

$$\min_{(j,s)} \left[\sum_{x_i \in R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{x_i \in R_2(j,s)} (y_i - \hat{y}_{R_2})^2 \right], \tag{3.1}$$

where $\hat{y}_{R_1}$ and $\hat{y}_{R_2}$ are given by $\hat{y}_{R_1} = mean(y_i|x_i \in R_1(j,s))$ and $\hat{y}_{R_2} = mean(y_i|x_i \in R_2(j,s))$.

Next, we divide one of the two resulting regions by repeating the above splitting process to minimize the RSS within the same region. The partition continues further on all of the resulting regions until some stopping criterion is reached.

The tree size is a measure of the complexity of the tree model. Large trees may overfit the data and result in large prediction error for the test data. Therefore, the optimal size of tree needs to be determined in order to improve the prediction performance of the model. One strategy is to stop splitting if the decrease in RSS for this split is less than some fixed threshold. This strategy may stop splitting too soon and omit some good splits on further steps. A better strategy is to grow a large tree until a minimum node size is reached first, and then prune it back by using cost-complex pruning.

We define the cost- complexity function as follows

$$R_\alpha(T) = R(T) + \alpha|\tilde{T}|, \tag{3.2}$$

where the regression cost

$$R(T) = \sum_{m=1}^{|\tilde{T}|} \sum_{i:x_i \in R_m} (y_i - \hat{y}_{R_m})^2$$

is RSS; α is a non-negative complexity parameter and can be considered as the complexity cost per terminal node; T is any sub-tree obtained by pruning the full tree T_0; $\tilde{T}$ is the set of terminal nodes of T and $|\tilde{T}|$ is the number of terminal nodes in T. Therefore $R_\alpha(T)$ is formed by adding a cost penalty for complexity to the RSS of the sub-tree.

If α is small, the penalty for having a large number of terminal nodes is small and the sub-tree which minimizes $R(T)$, T_α, will be large. For example, when $\alpha = 0$, T_α is the full tree T_0. As α increases, the penalty for having fewer terminal nodes is small and T_α will be small. If α is sufficiently large, the minimizing T_α will be the root node only.

For every value of α, one can show that there exists a unique smallest sub-tree T_α that minimizes R_α. Although α can run through a continuous range of values, the number of sub-trees is finite. Because of the finiteness, an optimal sub-tree is optimal for an interval range of α, and the number of such intervals has to be finite (Zhang and Singer, 2010). The limits of these intervals or thresholds of α and the smallest minimizing sub-tree can be found as follows.

Any internal node τ, has a number of offspring terminal nodes. The cost of the internal node and the offspring nodes can be denoted by $R(\tau)$ and $R(\tilde{T}_\tau)$, respectively, where $\tilde{T}_\tau$ contains the terminal nodes of T_τ and T_τ is a branch rooted at node τ. Then for each internal node τ, the value of α is given by

$$\alpha = \frac{R(\tau) - R(\tilde{T}_\tau)}{|\tilde{T}_\tau| - 1}. \tag{3.3}$$

We start with the smallest α_1 over the internal nodes. The corresponding node τ_1 is the weakest link in T_1, as α increases, it is the first node for which $R_{\alpha_1}(\tau_1)$ becomes equal to $R_{\alpha_1}(\tilde{T}_{\tau_1})$. Then we prune T_{τ_1}, namely treat the node τ_1 as a terminal node. This is so-called the weakest link pruning.

This pruning process results in the optimal sub-tree T_{α_1} corresponding to α_1. The interval for which the sub-tree T_{α_1} is optimal begins from α_1 and ends at the second threshold complexity parameter α_2. We choose the second threshold complexity parameter α_2 in the same way based on the pruned sub-tree T_{α_1}, and get the corresponding optimal sub-tree T_{α_2}.

Consequently, we create a decreasing sequence of sub-trees from a full tree to a root node,

$$T_{\alpha_0} > T_{\alpha_1} > T_{\alpha_2} > \cdots > \{root\ node\}$$

with a sequence of thresholds of complexity parameters α

$$0 = \alpha_0 < \alpha_1 < \alpha_2 \ldots$$

and the final optimal sub-tree can be selected from among them as follows.

For the final selection of sub-tree T_α with complexity parameter α, we need to have a good estimate for the regression costs of a sequence of sub-trees, which are

$$R(T_{\alpha_0}), R(T_{\alpha_1}), R(T_{\alpha_2}) \ldots, R(\{root\ node\})$$

with corresponding complexity parameters $0 = \alpha_0, \alpha_1, \alpha_2 \ldots$. They are estimated by using a k-fold cross validation (CV) process, in which the data set is randomly divided into k subsets. In this process, every subset is chosen as test data and the rest of the data is treated as training data. Thus, k pairs of training and test data sets are given. For each pair of data sets, by taking each complexity parameter α_i, $i = 1, 2 \ldots$ that we have already derived above, we use the training data set to build a sequence of optimal sub-trees in the previous way and use the test data set to calculate the prediction error for each sub-tree. Then, we get k estimates of $R(T_{\alpha_i})$, $i = 1, 2 \ldots$. By averaging the k estimates of $R(T_{\alpha_i})$, the estimate of each $R(T_{\alpha_i})$, $i = 1, 2 \ldots$ is obtained. The standard error of $R(T_{\alpha_i})$ can also be calculated from the k replicates.

The change of $R(T_{\alpha_i})$ with α_i can be demonstrated by a plot of the CV prediction error versus *cp* instead of α, where $cp = \alpha / R(\{root\ node\})$, the dimensionless complexity parameter. The final optimal sub-tree is the smallest sub-tree with cross validation error no larger than the minimum plus one standard error (1-SE). The 1-SE rule tends to yield a more robust and parsimonious model, as described by Breiman et al (1984).

3.2 Classification Trees

A classification tree is a model used to predict a qualitative response rather than a quantitative response. The process of building and pruning classification trees is very similar to that for regression trees. That is, we build classification trees by recursive binary partitioning and prune trees based on cost-complexity pruning. The only differences between these two processes are the splitting and pruning criteria. For a classification tree, the residual sum of squares is no longer the splitting criterion, instead we use misclassification error rate as the splitting criterion.

Suppose the response variable has K classes, so $y_i = k$, where $k = 1, 2, \ldots, K$. The proportion of class k observations in node m, is given by

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{i:x_i \in R_m} I(y_i = k), \tag{3.4}$$

where N_m is the number of observations in node m.

In this case, the estimated value of a response variable y_i in node m is the class with the largest proportion in node m.

The misclassification error rate in node m is the proportion of the observations that do not belong to the class with the largest proportion in node m. It can be expressed by

$$E_m = 1 - max(\hat{p}_{mk}). \tag{3.5}$$

The best classification tree is the one with the lowest misclassification error rate or impurity of its nodes. Hence, we are not only interested in the class prediction but also in the class proportion of a node. However, the misclassification error rate is sometimes not sensitive to impurity of its nodes. There are two other quantities to describe the impurity of a node. One is the Gini index and the other is the cross-entropy or deviance.

The Gini index is defined by

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}), \tag{3.6}$$

which measures the total variance of the response over the node m.

The cross-entropy or deviance is given by

$$D = \sum_{k=1}^{K} \hat{p}_{mk} log(\hat{p}_{mk}), \tag{3.7}$$

and is an alternative to the Gini index.

Since the Gini index and deviance are more sensitive than the misclassification error rate to the impurity of a node (Hastie et al, 2009), we use the Gini index to grow a classification tree and use the misclassification error rate for pruning the tree in order to achieve an optimal size of a classification tree.

3.3 Results

3.3.1 SGA

We start by growing a large tree, then prune it back using the dimensionless complexity parameter (*cp*) pruning technique to achieve an optimal size tree. The **rpart** function in R is used on the training data set. The Gini index is used in tree splitting. Since the observed prevalence of SGA is very low (0.08) and trees will not accurately predict the minority class when applied to class-imbalanced data, weights of 10 are applied to the second class of sga ($sga = 1$). The value of *cp* is chosen at 0.002, meaning that the classification tree will be grown until $cp = 0.002$ is achieved. This initial tree for SGA is shown in Figure 3.1. The class membership and sample composition are displayed. Inside each node is the fitted class of *sga*. Under each node are the number of the first class of *sga* (left) and the number of second class of *sga* multiplied by 10 (right). The root node is classified as $sga = 0$ because there are more non SGA babies (41776) than SGA babies (3204) multiplied by 10 (32040). The first split is on *smk*, with nonsmokers classified as $sga = 0$ and the remaining categories (< 0.5 pack/day, $>= 0.5$ pack/day and Unknown) classified as $sga = 1$.

In the tree pruning process, the misclassification costs of a sequence of sub-trees with corresponding *cp* are estimated using 10-fold cross validation. The Table 3.1 lists the *cp*, the number of splits (*nsplit*), the relative error (*rel error*), the relative cross validation error (*xerror*), and the standard error of the relative cross validation error (*xstd*) for the initial SGA tree. The relative error is defined as the misclassification

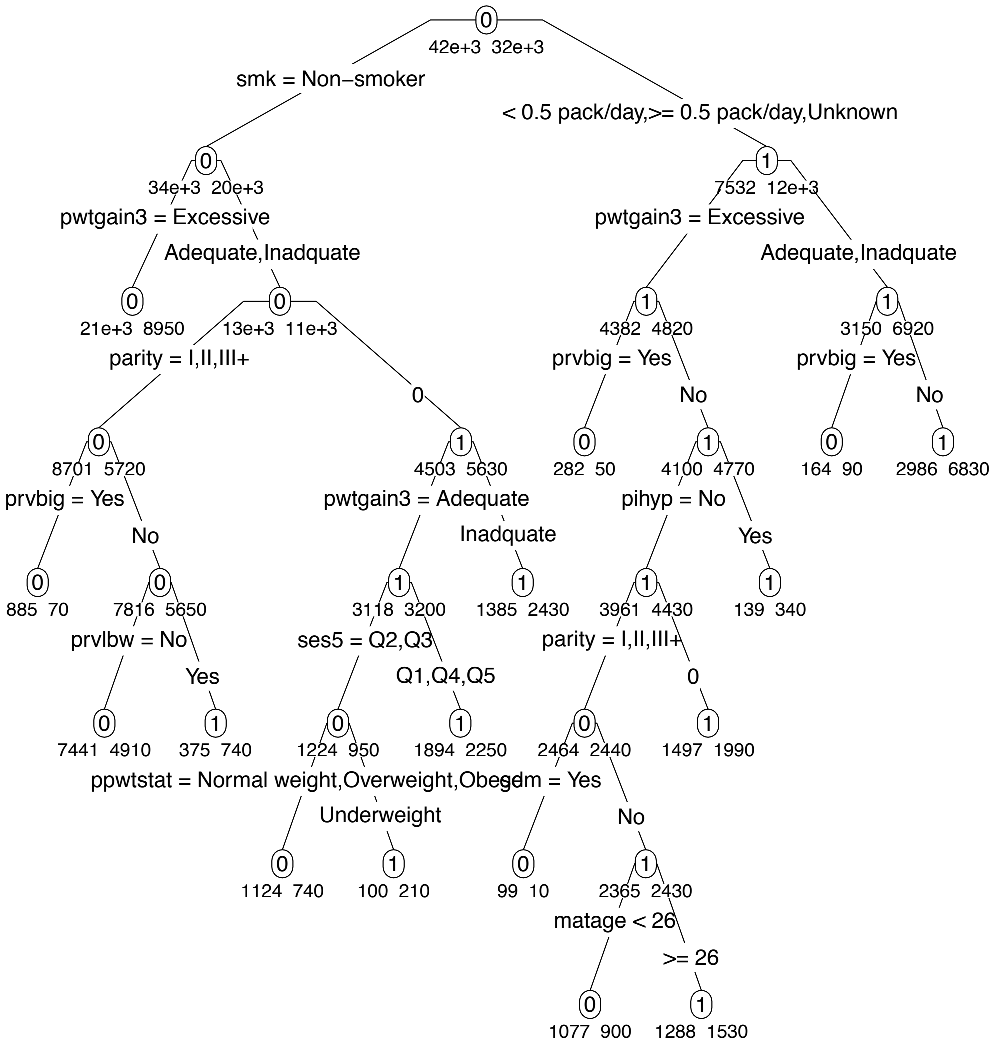

Figure 3.1: The initial tree for SGA. Inside each node is the fitted class of *sga*. Under each node are the number of $sga = 0$ (left) and $(sga = 1) \times 10$ (right).

error divided by that of the root node. The graphical presentation of relative cross validation error versus complexity parameter and tree size (always 1 + number of split) is given in Figure 3.2.

cp	*nsplit*	*rel error*	*xerror*	*xstd*
0.1313	0	1	1	0.0042
0.0176	1	0.8687	0.8687	0.0041
0.0057	3	0.8335	0.8297	0.0041
0.0043	5	0.8221	0.8254	0.0041
0.0036	7	0.8135	0.8194	0.0041
0.0034	9	0.8063	0.8122	0.0041
0.0023	10	0.8029	0.8069	0.0040
0.0021	11	0.8006	0.8052	0.0040
0.0020	15	0.7923	0.8052	0.0040

Table 3.1: Complexity parameter table for the initial SGA tree. The complexity parameter (*cp*), the number of splits (*nsplit*), the relative error (*rel error*), the relative cross validation error (*xerror*), and the standard error of the relative cross validation error (*xstd*) for the initial SGA tree are given.

Both the numerical output and the plot indicate that the minimal cross validation error, 0.8052, was reached with a cross validation error standard error of 0.0040 when the tree has 12 terminal nodes or 11 splits. Using the 1-SE rule, the smallest subtree is found with the error below 0.8052 +0.0040 = 0.8092. This leads to the pruned tree with 11 terminal nodes or 10 splits as displayed in Figure 3.3 and Figure 3.4.

Both Figure 3.3 and Figure 3.4 show the pruned tree with splitting rules and the fitted class of *sga*. The former presents the number of sga=0 (left) and the number $10 \times (sga = 1)$ for each node, and the latter presents the probability of $sga = 1$ given that node. The pruned tree model is constructed by the following predictors: *smk*, *pwtgain*3, *parity*, *pribig*, *ppwtstat*, *prelbw* and *ses5* from the root (top) to leaves (bottom), suggesting that they are very important variables for predicting SGA births. Normally the higher the variable occurs in the tree, the higher the predictive power and importance of the variable.

The first split is on *smk*, which is the most important predictor. Non-smokers are assigned to the left branch with $sga = 0$, while categories $smk < 0.5$ pack/day, *smk* >= 0.5 pack/day and *smk*Unknown are assigned to the right branch with $sga = 1$. This indicates that women who smoke or whose smoking status is unknown have a

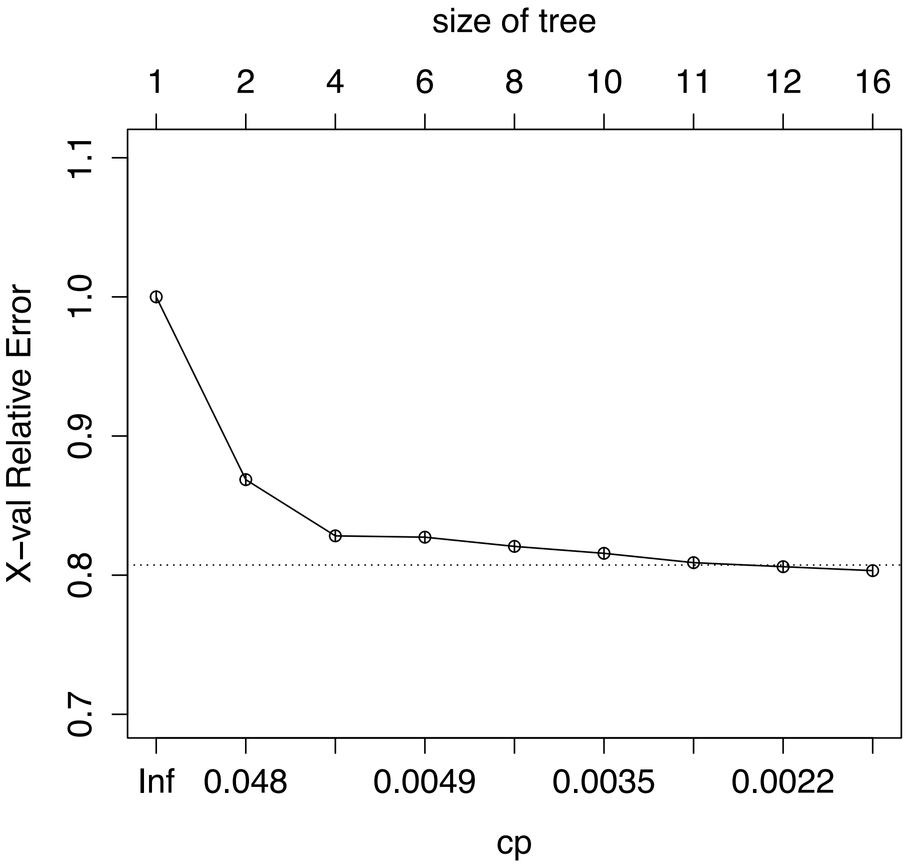

Figure 3.2: The relative cross validation error change with the complexity parameter and tree size for the initial SGA tree.

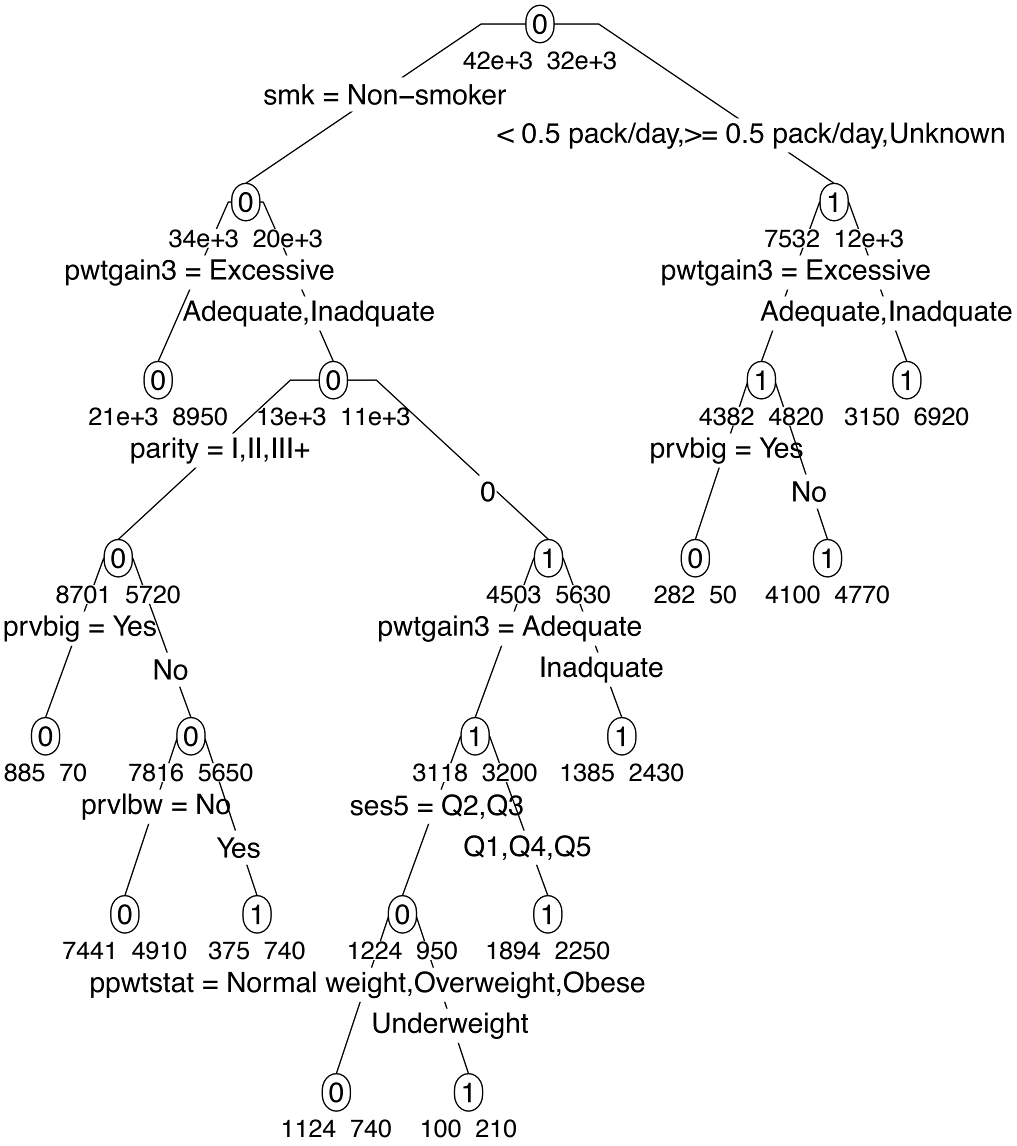

Figure 3.3: The pruned tree for SGA. Inside each node is the fitted class of sga. Under each node are the number of $sga = 0$ (left) and $(sga = 1) \times 10$ (right), respectively.

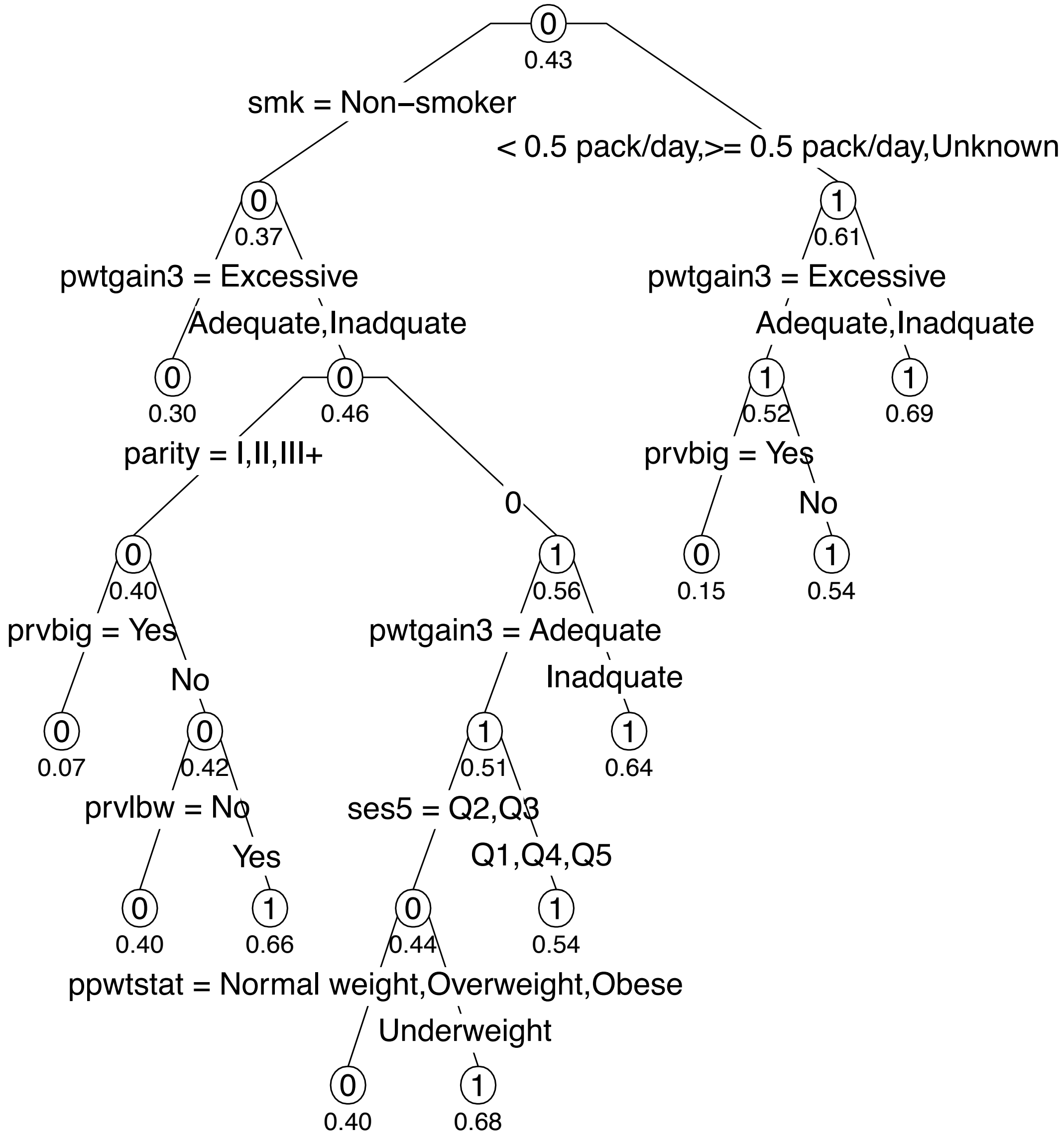

Figure 3.4: The pruned tree for SGA. Inside each node is the fitted class of *sga*. Under each node is the probability of $sga = 1$ with the number of $(sga = 1) \times 10$ given the node.

higher odds to deliver SGA babies than those who do not smoke.

Following the right branch of the first split, this group is further subdivided by *pwtgain*3, which is the second most important factor for predicting SGA births. Mothers who have excessive pregnancy weight gain have a lower probability of delivering SGA babies than those who have adequate or inadequate weight gain. *prvbig* is a negative predictor of SGA births. Mothers who have previous birth weights larger than 4080 g have a lower odds of delivering SGA babies than those who have not had a previous baby larger than 4080 g birth weight.

Following the left branch of the first split on *smk*, this group is also further subdivided by *pwtgain*3, suggesting that mothers with both adequate or inadequate weight gain have a higher risk of delivering SGA babies than those with excessive pregnancy weight gain. The former group of women is further split by *parity*: Multiparous women have a lower risk of delivering SGA babies than nulliparous women.

Along the right branch of splitting node on *parity*, this group continues to be separated into cases with inadequate and adequate pregnancy weight gain. Those who have inadequate pregnancy weight gain have a higher probability of delivering SGA babies than those who have adequate weight gain. The cases with adequate weight gain is split by *ses5* (income quintile based on the woman's residence census dissemination area), showing that women from Q1, Q4, and Q5s tend to have higher probability of delivering SGA babies than those from Q2 and Q3. The group of women from Q2 and Q3 are further split by*ppwtstat*, indicating that mothers with underweight pre-pregnancy weight status have higher risk of having SGA babies than those of normal weight, overweight and obese pregnancy weight status.

Along the left branch of the splitting node on *parity*, the group of multiparous women is further split by *prvbig*: mothers who have previous babies larger than 4080 g have much lower risk of giving SGA births than those who have not had previous babies larger than 4080 g birth weight. This node is further split on *prvlbw*: women who had a previous baby with a birth weight less than 2500g birth have a higher risk of delivering an SGA infant than those who did not have a baby with birth weight less than 2500g .

Figure 3.5 illustrates the probability of each observation in each node relative to all the observations. Inside each node is the class of *sga*. Under each node, the left

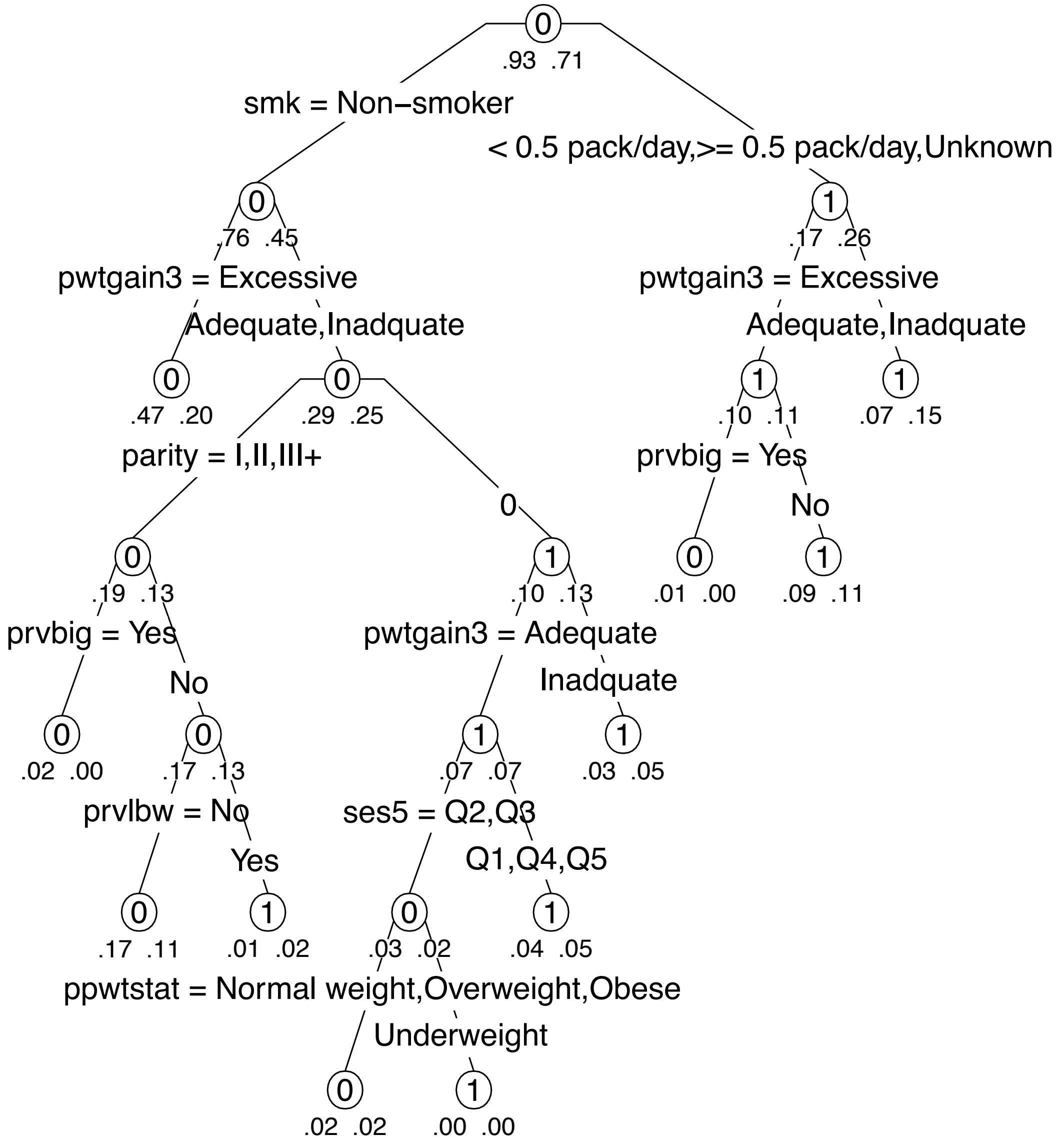

Figure 3.5: The pruned tree for SGA. Inside each node is the class of *sga*. Under each node are the probability relative to all observations for the first class of $sga = 0$ (left) and for the second class of $sga = 1$ with a weight of 10 (right).

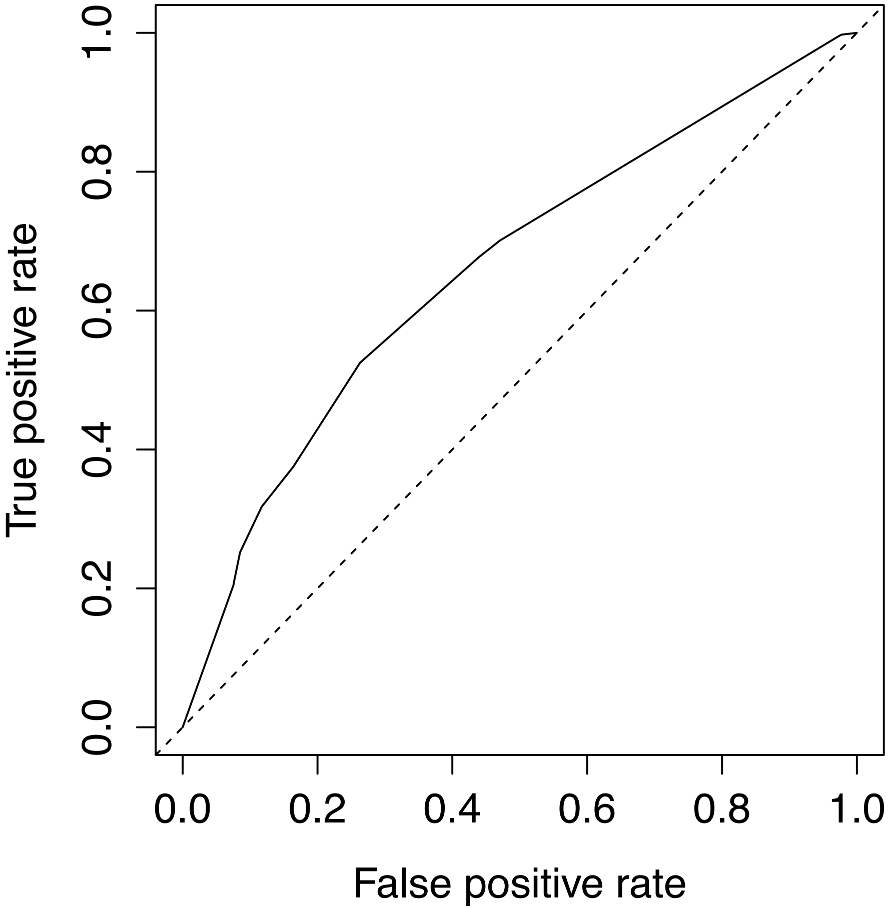

Figure 3.6: The ROC curve for the pruned tree for SGA.

number is the probability of *sga* = 0, while the right number is the probability of *sga* =1. We should notice that the actual probability for the second class of *sga* in each node is the right number divided by the weight of 10. This figure provides more accurate results about the probability of *sga* class in each node for future observations.

The ROC curve of the classification tree model for SGA on the test data is plotted in Figure 3.6. Because the curve is above the line of equality, this ROC curve demonstrates that the classification tree model improves the prediction precision over a random prediction model. The area under the ROC curve is 0.6603, which is smaller than that obtained using logistic regression.

The ROC curve is not as smooth as that for logistic regression. For a tree model with only a few terminal nodes, the threshold only changes when the probability in a terminal node is reached. For the logistic model, a fitted or predicted probability is obtained for each case, so the probability threshold can be changed in small steps to trace out a smooth ROC curve.

The optimal threshold derived from the ROC curve is 0.538, even larger than 0.5. With the optimal threshold, the sensitivity is expected to be even lower than that for probability threshold of 0.5. Therefore, the threshold probability of 0.5 is selected in the diagnostic test for the SGA classification tree model using test data.

The confusion matrix and diagnostic summaries are shown in Table 3.2.

	Disease+	Disease-	Total
Test+	200	1187	1387
Test-	181	3324	3505
Total	381	4511	4892

Point estimates and 95% CIs		
Apparent prevalence	0.28	(0.27,0.30)
True prevalence	0.08	(0.07,0.09)
Sensitivity	0.52	(0.47,0.58)
Specificity	0.74	(0.72,0.75)
Positive predictive value	0.14	(0.13,0.16)
Negative predictive value	0.95	(0.94,0.96)
Positive likelihood ratio	1.99	(1.79,2.22)
Negative likelihood ratio	0.64	(0.58,0.72)

Table 3.2: The confusion matrix and diagnostic summaries for the SGA tree with threshold 0.5.

The confusion matrix reveals that the classification tree model predicted that 1387 people would deliver SGA babies. Of the 381 individuals who actually deliver SGA babies, the tree model correctly predicts 200, or 52% (sensitivity). 1187 out of 4511 of the individuals who did not deliver SGA babies were incorrectly labeled, so the specificity is 74%. The overall prediction error is 27.96%.

The positive likelihood ratio (1.99) and the negative likelihood ratio (0.64) indicate that a woman with an SGA birth is about 1.99 times as likely to have a positive test than a women without a SGA birth, and the probability of having a negative test for women with a SGA birth is 0.64 of that of those without a SGA birth .

3.3.2 LGA

We use the same approach to fit a classification tree for LGA as for SGA. We start by growing a large tree, and then prune it back using the dimensionless complexity parameter (*cp*) pruning technique to achieve an optimal size of tree. The classification tree for LGA is built using the **rpart** function on the training data. The Gini index is used in tree splitting. Since the observed prevalence of LGA is very low (0.15), the weights of 5 are applied to the cases with $lga = 1$. The value of *cp* is chosen as 0.002, so that the classification tree is grown until $cp = 0.002$ is reached. This initial tree is shown in Figure 3.7. The class membership and sample composition are displayed. Inside each node is the *lga* fitted class. Under each node are the number of $lga = 0$ cases (left) and the number of second class of $lga = 1$ cases multiplied by 5 (right).

In the tree pruning process, the misclassification costs of a sequence of sub-trees with corresponding values of *cp* are estimated based on 10-fold cross validation. Table 3.3 lists the *cp*, the number of splits (*nsplit*), the relative error ($relerror$), the relative cross validation error ($xerror$) and the standard error of the relative cross validation error($xstd$) for the initial LGA tree. The graphical presentation of relative cross validation error versus complexity parameter and tree size (always 1 + number of splits) is given in Figure 3.8.

cp	*nsplit*	*rel error*	$xerror$	$xstd$
0.1503	0	1.000	1.000	0.0040
0.019	1	0.8497	0.8497	0.0039
0.0077	4	0.7926	0.7901	0.0038
0.0072	6	0.7772	0.7887	0.0038
0.002	7	0.7680	0.7781	0.0038

Table 3.3: Complexity parameter table for the initial LGA tree. The complexity parameter *cp* , the number of splits (*nsplit*), the relative error (*rel error*), the relative cross validation error ($xerror$) and the standard error of the relative cross validation error($xstd$) for the initial LGA tree.

Both the numerical output and the plot indicate that the minimal error, 0.7781, was reached with a standard error 0.0038 when the tree has 8 terminal nodes or 7 splits. By using the 1-SE rule, the smallest subtree is found with the error below $0.7781 + 0.0038 = 0.7819$. This indicates that the initial LGA tree is the optimal tree

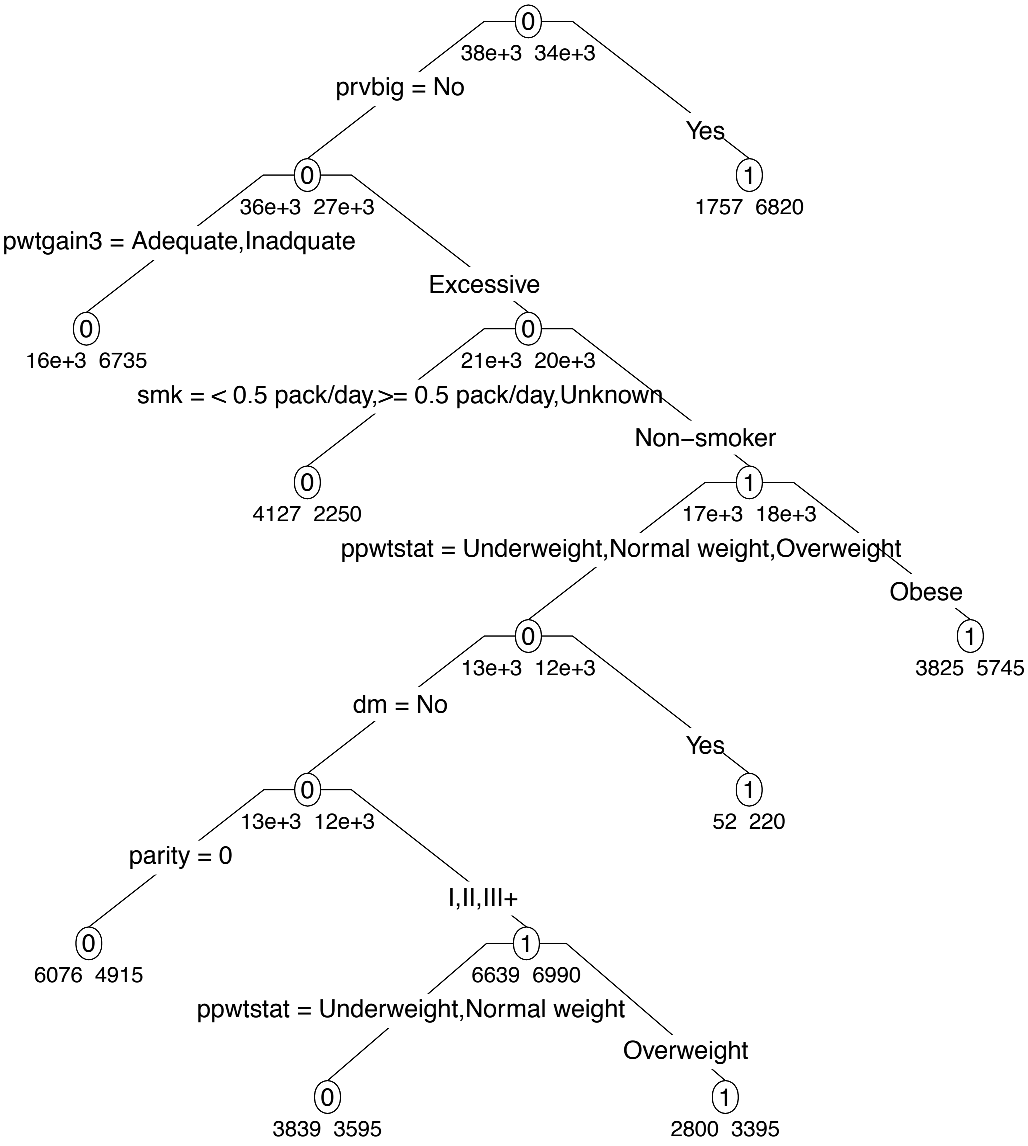

Figure 3.7: The initial tree for LGA. Inside each node is the class of *lga*. Under each node are the number of $lga = 0$ (left) and $(lga = 1) \times 5$ (right).

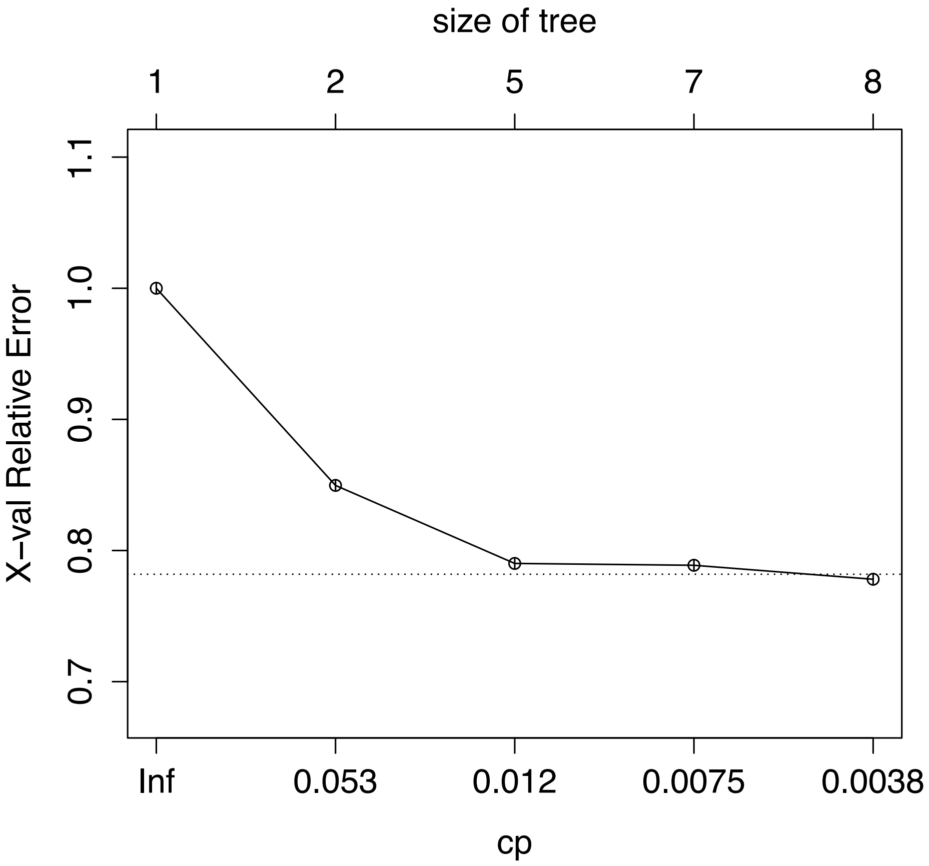

Figure 3.8: The relative cross validation error change with the complexity parameter for the initial LGA tree.

without need of pruning. The probability of *lga* =1 in each node is shown in Figure 3.9.

Figure 3.9 suggests that *prvbig*, *pwtgain*3, *smk*, *ppwtstat*, *dm* and *parity*, from the root (top) to leaves (bottom) are important predictors of LGA birth in descending importance.

The first split is on *prvbig*, which is the most important predictor. Women who had a previous birth with a birth weight larger than 4080 *g* are more likely to deliver an LGA infant than those who did not previously have a baby larger than 4080 *g* birth weight.

The women who had previous births with birth weights less than 4080 *g* is further subdivided by *pwtgain3*, indicating that *pwtagain3* is the second most important factor to predict LGA birth. Mothers who have excess pregnancy weight gain have a higher probability of having LGA babies than those who have adequate or inadequate weight gain. The former group is further split by *smk*: the nonsmoker group tends to have a higher risk of LGA birth than the smokers or those with unknown smoking status. This group is then separated by *ppwstat*: mothers with an obese pre-pregnancy weight status have a higher risk of having LGA babies than those who are underweight, normal weight or overweight. The latter group is divided by *dm*: women with pre-existing diabetes have higher probability of delivering LGA babies than those without pre-existing diabetes. The group of women without pre-existing diabetes are split by *parity*. This demonstrates that multiparous women have a slightly higher risk of delivering LGA babies than nulliparous women. These cases are further divided by *ppwstat*, suggesting that mothers with overweight pre-pregnancy weight status have higher risk of LGA births than those who have underweight or normal weight pre-pregnancy weight status.

Figure 3.10 illustrates the probability of each observation in each node relative to all the observations. Inside each node is the *lga* class. Under each node, the left number is the probability of *lga*=0, while the right number is the probability of *lga*=1. We should notice that the real probability for *lga*=1 in each node is the right number divided by the weight 5. This figure provides more accurate results about the probability of *lga* class in each node for future observations.

The ROC curve of the classification tree model for LGA on the test data is plotted

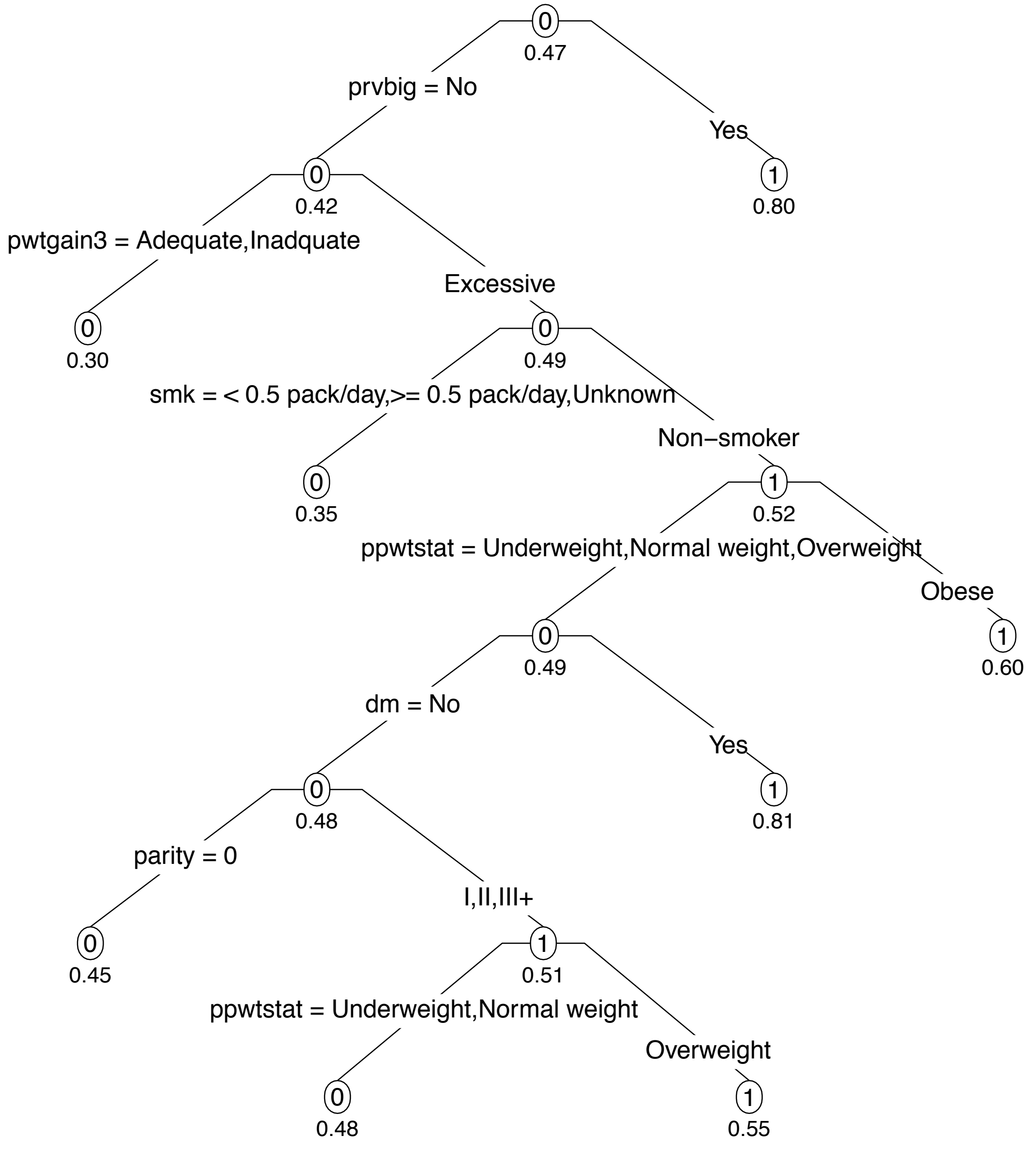
0
0.47
prvbig = No
Yes
1
0.80
0
0.42
pwtgain3 = Adequate,Inadquate
Excessive
0
0.30
0
0.49
smk = < 0.5 pack/day,>= 0.5 pack/day,Unknown
Non−smoker
0
0.35
1
0.52
ppwtstat = Underweight,Normal weight,Overweight
Obese
1
0.60
0
0.49
dm = No
Yes
1
0.81
0
0.48
parity = 0
I,II,III+
0
0.45
1
0.51
ppwtstat = Underweight,Normal weight
Overweight
0
0.48
1
0.55

Figure 3.9: The optimal tree for LGA. Inside each node is the class of *lga*. Under each node are the probability of $lga = 1$ with the number of $lga = 1 \times 5$ given the node.

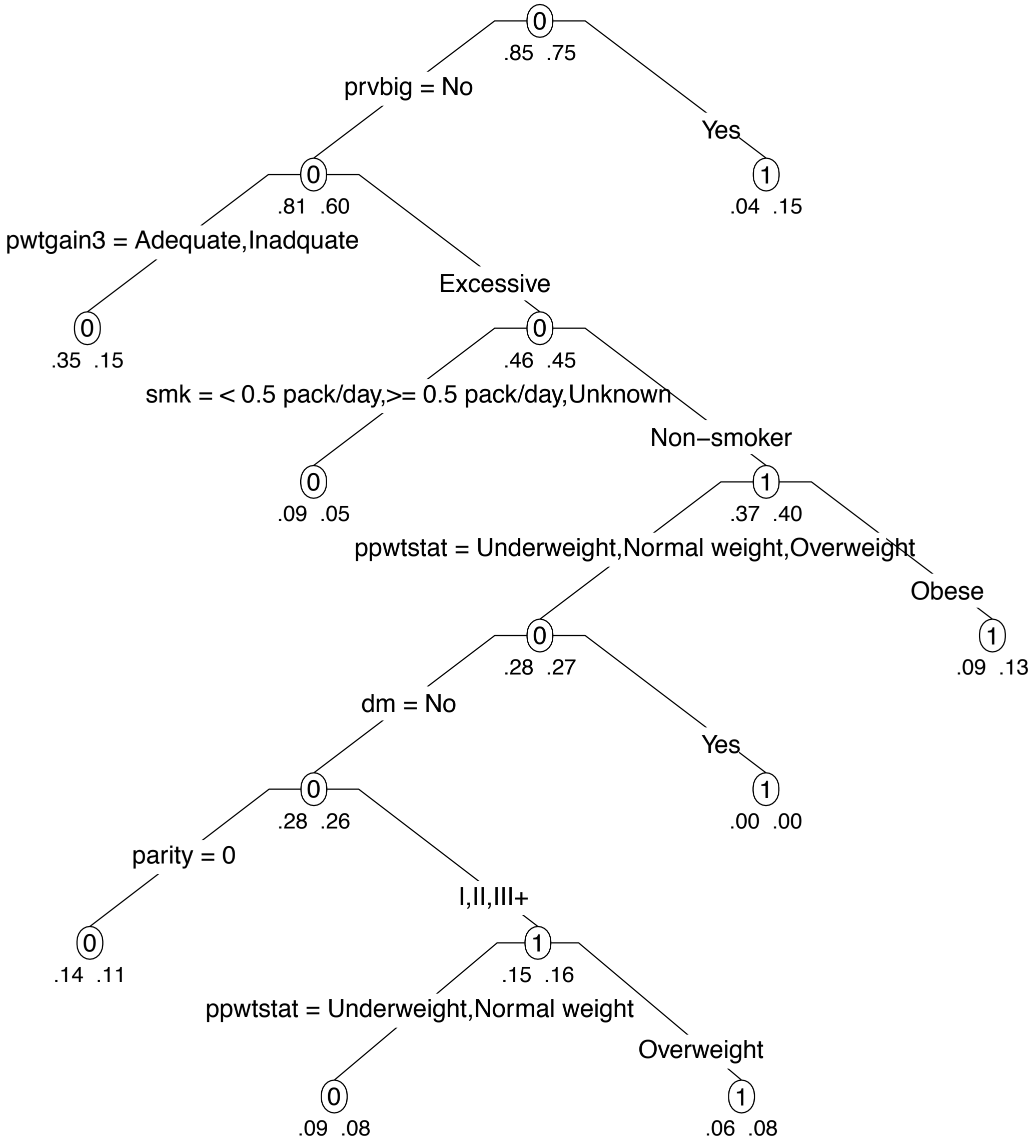

Figure 3.10: The optimal tree for LGA. Inside each node is the class of *lga*. Under each node is the probability relative to all observations for *lga* = 0 (left) and for *lga* = 1 the weight of 5 (right) in the node.

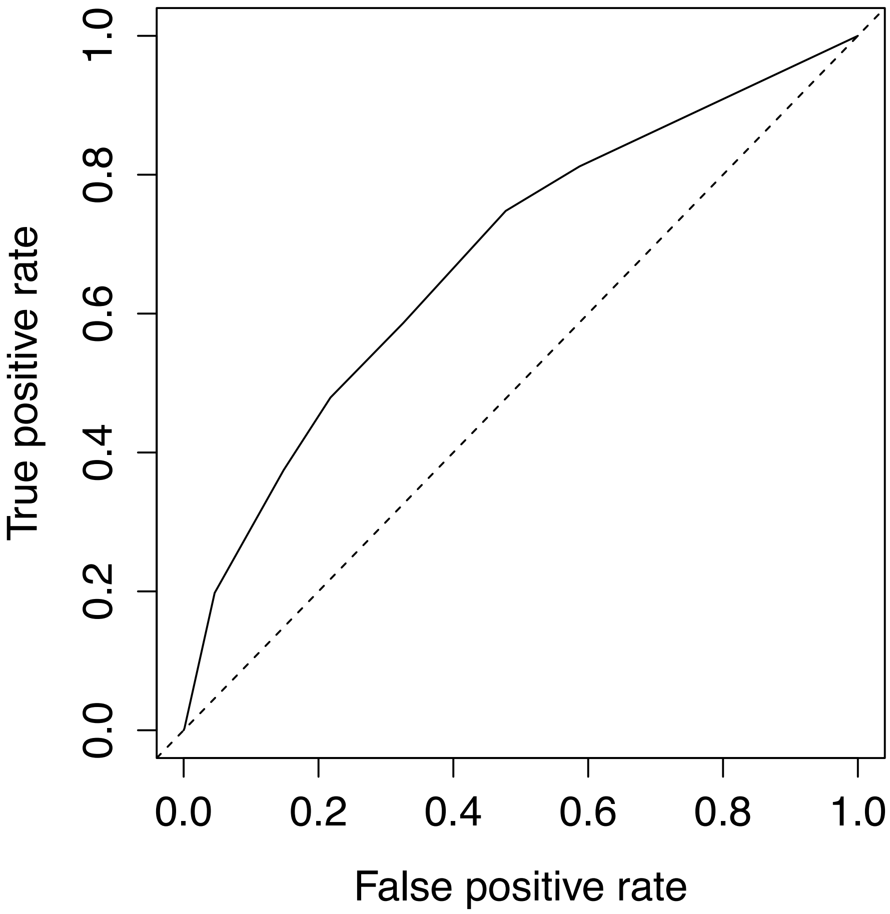

Figure 3.11: The ROC curve of optimal LGA tree.

in Figure 3.11. Because the curve is above the line of equality, this ROC curve demonstrates that the classification tree model improves the prediction precision over a random prediction model. The area under the ROC curve is 0.6817, which is a little lower than that obtained using logistic regression.

The optimal threshold derived from the ROC curve is 0.484. The diagnostic properties of the classification rule using this threshold are shown in Table 3.4.

With the optimal threshold, the classification model predicts that 1780 mothers will deliver LGA babies. Of the 718 individuals who deliver LGA babies, the classification model correctly predicts 421, or 59% (sensitivity). 1359 individuals who are

	Disease+	Disease-	Total
Test+	421	1359	1780
Test-	297	2815	3112
Total	718	4174	4892

Point estimates and 95% CIs		
Apparent prevalence	0.36	(0.35,0.38)
Ture prevalence	0.15	(0.14,0.16)
Sensitivity	0.59	(0.55,0.62)
Specificity	0.67	(0.66,0.69)
Positive predictive value	0.24	(0.22,0.26)
Negative predictive value	0.90	(0.89,0.91)
Positive likelihood ratio	1.80	(1.67,1.94)
Negative likelihood ratio	0.61	(0.56,0.67)

Table 3.4: The confusion matrix and diagnostic summaries for LGA tree with threshold 0.484.

not $lga = 1$ class are incorrectly classified. As a result, the overall error rate is 0.3385.

The positive likelihood ratio (1.80) and the negative likelihood ratio (0.61) reveal that a woman with a LGA baby is about 1.80 times more likely to have a positive test than a mother without a LGA baby, and the probability of having a negative test for women with LGA babies is 0.61 of that of those without LGA babies.

Compared with the logistic regression model, the tree model has several advantages. It has a more intuitive graphical presentation and can be easily explained. It is easier to handle interactions without the need for creating interaction terms. However, the tree model has overall lower predictive accuracy with lower sensitivity and smaller area under ROC curve. In the next chapters, other tree-based techniques will be introduced and applied in order to improve the predictive performance of the trees.

Chapter 4

Random Forests

The predictive accuracy of a tree model can be significantly improved by an ensemble of tree models, in which each tree is grown from a set of bootstrapped training data. Prediction is made by averaging the prediction from each tree for a continuous response or by the most popular class for a categorical response. This method, called bootstrap aggregating or bagging, can reduce the high variance of a tree model. It was first introduced by Breiman (1996) and became the precursor of the random forest model (Breiman, 2001).

Dietterich (1998) proposed random split selection where at each node the split is selected at random from several of the best splits and found that this approach does better than bagging. In bagging, the trees are highly correlated because they are grown by choosing almost the same important predictors at the top node. Thus bagging may not substantially reduce the variance over a single tree (James et al, 2013). The random forest approach de-correlates the trees in the collection by randomly selecting a small fraction of p predictors at each split, and provides an improvement over bagging in decreasing the variance of the model.

In this chapter, the random forest methodology and some important features are introduced. Then random forests are applied to the NSAPD data set to predict SGA and LGA.

4.1 Theory

Following Hastie et al (2009), in bagging, B bootstrapped training sets are drawn with replacement from the original training data set and trees are grown on each sample without pruning. This ensemble of trees can be denoted by $\hat{f}^{*1}(\mathbf{x}), \hat{f}^{*2}(\mathbf{x}), \ldots, \hat{f}^{*B}(\mathbf{x})$. For regression trees, the fitted model can be expressed as

$$\hat{f}_{bag}(\mathbf{x}) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^{*b}(\mathbf{x}). \tag{4.1}$$

For classification tree, the class predicted by each of the B trees is recorded and a majority vote is taken.

If the set of trees are identically distributed with variance σ^2 and positive pairwise correlation ρ, the variance of the bagging prediction $\hat{f}_{bag}(\mathbf{x})$ is

$$Var(\hat{f}_{bag}(\mathbf{x})) = \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2, \tag{4.2}$$

where the second item approaches zero if B is large enough. However, ρ is usually large for bagging because of the similarity of the top structure of each tree, so the first part of variance for bagging cannot be reduced by increasing the number of trees.

The random forest approach aims to reduce the first part of the variance for bagging by de-correlating trees in the ensemble. After B bootstrapped training data sets are drawn, each random-forest tree T_b, $b = 1, \ldots, B$ is grown to each of the data sets. At each node, instead of using the full set of p predictors as splitting variable candidates, m variables are randomly chosen from the p variables, and the best variable and split-point are chosen from the m variables. The random forest prediction at a new point $\mathbf{x}$ can then be expressed by

$$\hat{f}_{rf}^{B}(\mathbf{x}) = \frac{1}{B}\textstyle\sum_{b=1}^{B} T_b(\mathbf{x}) \tag{4.3}$$

for regression, and

$$\hat{C}_{rf}^{B} = majority\ vote\{\hat{C}_b(\mathbf{x})\}_1^B \tag{4.4}$$

for classifications, where $\hat{C}_b(\mathbf{x})$ is the class prediction of the b^{th} random-forest tree.

The trees in a random forest grown in this way are much less correlated, so that the variance of a random forest is much smaller than for a single tree. In general, the variance of a random forest prediction decreases as m, the number of variables considered at each split, decreases. The typical value of m is selected as $m = \sqrt{p}$ for a classification random forest and $m = p/3$ for a regression random forest.

The use of out-of-bag (OOB) samples is an important feature of random forest methodology. For any observation $(y_i, \mathbf{x}_i)$ in the original training data set, approximately two-thirds of bootstrapped samples include $(y_i, \mathbf{x}_i)$. The remaining one-third samples not including $(y_i, \mathbf{x}_i$ are referred to as OOB samples for observation $(y_i, \mathbf{x}_i)$. This can be argued as follows:

Each observation has probability $1/n$ of being selected for the bootstrap sample at each of n independent selections. The probability of the observation being selected is therefore binomial with index n and probability $1/n$. The probability of the observation not being selected is $(1 - \frac{1}{n})^n$. For large n this is approximately $1/e = 0.37$.

For each observation in the training data, its random forest prediction is made using only those trees constructed without this observation. Thereby, the OOB mean squared error or classification error can be determined and combined over all observations. Breiman (2001) pointed out that the OOB error estimate is as accurate as using a test set of the same size as the training set. In practice, the OOB error can be monitored as trees are added to the random forest. The training can be terminated once the OOB error has settled down and in this way the number of random forest trees B is determined.

OOB samples can also be used in two ways to measure the importance of variables. As Breiman (2001) introduced, for the b^{th} classification tree in a random forest, the OOB cases are put down the tree, and the number of votes for the correct class is counted. Then, the values of variable x_k among the OOB samples are randomly permuted and the OOB cases with the permuted values of variable x_k are put down the tree and the number of votes for the correct class are recounted. The same process is repeated for other trees in the forest. The difference between the number of votes for the correct class in the original OOB data and in the variable-x_k-permuted OOB data averaged over all B trees in the forest is used to measure the importance of variable x_k. This is called variable permutation (or randomization) importance.

Another variable importance measure for classification trees in a random forest is Gini importance. The Gini criterion in construction of a classification tree is used as the splitting criterion for each single classification tree in the random forest. The Gini importance of variable x_k is measured by adding up the decrease in the Gini index by splits over variable x_k, averaged over all B trees.

The variable importance measures for classification trees in random forest can be extended to regression trees in random forests. The permutation importance of a variable x_k is the average increase in mean squared error (MSE) resulting from the random permutation of values of x_k in OOB samples over all B trees in the forest. Another variable importance measure for regression trees in a random forest is obtained by accumulating the decrease in MSE by splits over variable x_k and averaged over all B trees.

A random forest model provides improvement of the prediction accuracy over a tree model at the expense of interpretability and intuitive graphical presentation. The random forest structure cannot be presented graphically like the tree model. However, the variable importance measure gives the summarized and quantitative information of the random forest. It can be used for variable selection and the important predictors can further be used by other models.

4.2 Results

4.2.1 SGA

A classification random forest was grown on the training data for SGA using the **randomForest** package in R. As was done with the classification tree model, weights of 10 are applied to the second class of *sga* (*sga*=1). The size of the set of randomly selected predictor variables used for determining each binary split is $m = \sqrt{p}$, where p denotes the total number of predictor variables. In our study, $p = 17$ and so $m = 4$. 500 classification trees were grown for the random forest.

Figure 4.1 shows the OOB misclassification error progression on 500 trees. The black line demonstrates the OOB error over all classes as a function of the number of trees in the model. The OOB misclassification error for the first class of *sga* (*sga*=0) (red dashed line) is lower than that of the second class of *sga* (*sga*=1) (green dotted line). The OOB error is stabilized at 500 trees. The overall OOB misclassification error for the SGA random forest with 500 trees is 0.2378, which is much smaller than the cross validation error of the single classification tree from Chapter 3 (0.8052). The OOB misclassification errors for $sga = 0$ and $sga = 1$ are 0.1735 and 0.3216 respectively.

Figure 4.2 displays the rankings of variable importance for the SGA random forest.

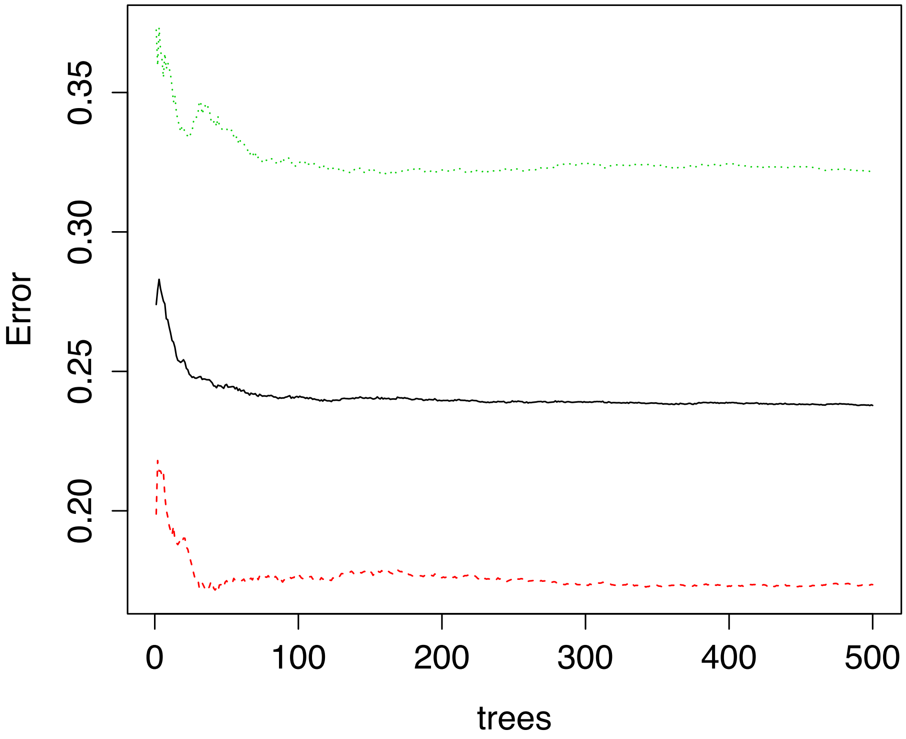

Figure 4.1: Random forest results for SGA. The OOB misclassification error (black line) is shown as a function of the number of trees. The number of predictors used for splitting at each node is $m = 4$. The green dotted line and green dashed lines represent the OOB error for $sga = 1$ and $sga = 0$ respectively.

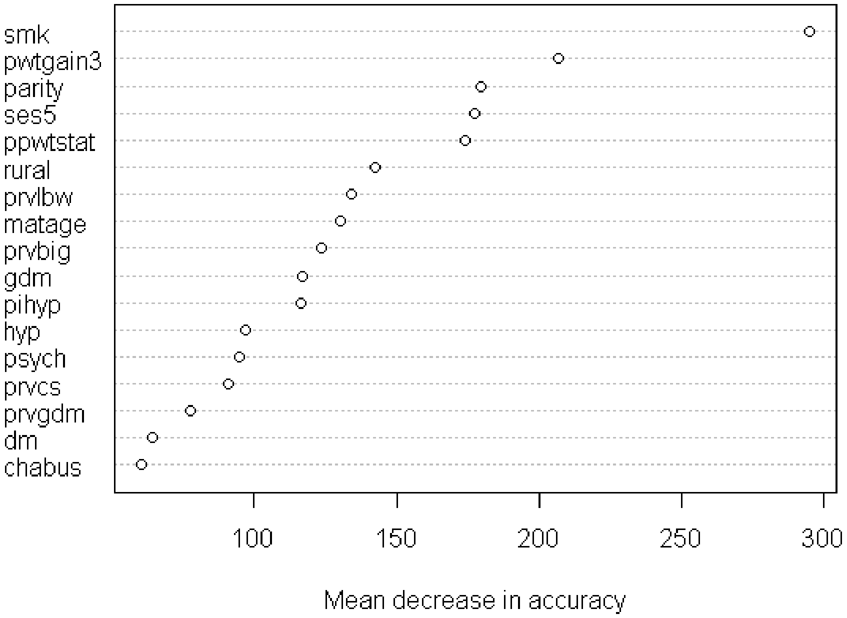

Figure 4.2: A variable importance plot for a SGA random forest. Variable importance is computed using the mean decrease in accuracy over all classes for OOB permutation.

Variable importance was calculated using the permutation approach. The top two influential predictors are *smk* and *pwtgain*3, followed by *parity*, *ses*5 and *ppwtstat*.

The ROC curve of the random forest model for SGA on the test data is plotted in Figure 4.3. Because the curve is above the line of equality, this ROC curve demonstrates that the random forest model improves the prediction precision over a random prediction model. The area under the ROC curve is 0.6752, which is slightly larger than that obtained using a tree model (0.6603).

The optimal threshold calculated from the ROC curve is 0.276. The diagnostic properties of the classification rule using this threshold are shown in Table 4.1.

With the optimal threshold, the classification model predicts that 1672 individuals will deliver SGA babies. Of the 381 individuals who deliver SGA babies, the classification model correctly predicts 228, or 60% (sensitivity). 1444 individuals who are not $sga = 1$ class are incorrectly classified. As a result, the overall error rate is 32.65%.

The positive likelihood ratio (1.87) and the negative likelihood ratio (0.59) indicate

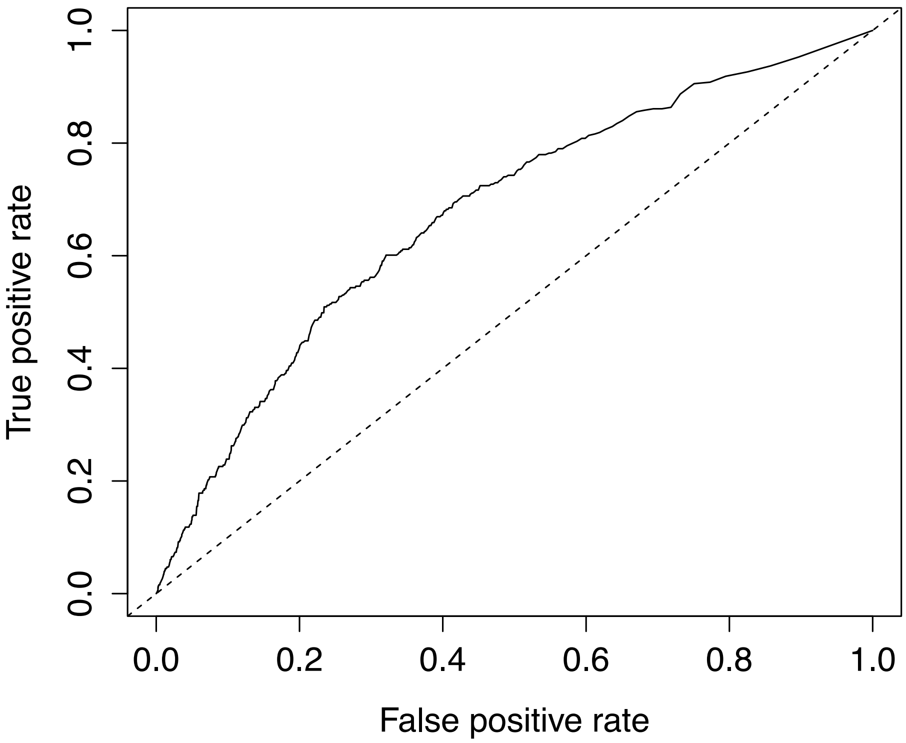

Figure 4.3: The ROC curve of the SGA random forest.

	Disease+	Disease-	Total
Test+	228	1444	1672
Test-	153	3067	3220
Total	381	4511	4892

Point estimates and 95% CIs		
Apparent prevalence	0.34	(0.33, 0.36)
True prevalence	0.08	(0.07, 0.09)
Sensitivity	0.60	(0.55, 0.65)
Specificity	0.68	(0.67, 0.69)
Positive predictive value	0.14	(0.12, 0.15)
Negative predictive value	0.95	(0.94, 0.96)
Positive likelihood ratio	1.87	(1.70, 2.05)
Negative likelihood ratio	0.59	(0.52, 0.67)

Table 4.1: The confusion matrix and diagnostic summaries for the SGA random forest with threshold 0.276.

that a mother with a SGA baby is about 1.87 times more likely to have a positive test than a mother without a SGA baby, and the probability of having a negative test for mothers with SGA babies is 0.59 of that of those without SGA babies.

4.2.2 LGA

A classification random forest was grown on the training data set using the **randomForest** package in R. As was done with the classification tree model for LGA, weights of 5 are applied to the second class of *lga* (*lga*=1). The size of the set of randomly selected predictor variables used for determining each binary split is $m = \sqrt{p}$, where p denotes the total number of predictor variables. In our study, $p = 17$ and so $m = 4$. 500 classification trees were grown for the random forest.

Figure 4.4 shows the OOB misclassification error progression on 500 trees. The black line demonstrates the OOB error over all classes as a function of the number of trees in model. The OOB misclassification error for the first class of *lga* (*lga*=0) (red dashed line) is lower than that of the second class of *lga* (*lga*=1) (green dotted line). It appears that the OOB error estimate is stabilized at 500 trees. The overall OOB misclassification error for the LGA random forest with 500 trees is 0.2981, which is much smaller than the cross validation error of the single classification tree from Chapter 3 (0.7781). The OOB misclassification errors for lga = 0 and lga = 1 are 0.2458 and 0.3575 respectively.

Figure 4.5 displays the rankings of variable importance for LGA random forest. Variable importance is calculated using the permutation approach. The top two influential predictors are *smk* and *prvbig*, followed by *pwtgain*3, and *ppwtstat. ses*5 is also in the higher rank.

The ROC curve of the random forest model for LGA on the testing data is plotted in Figure 4.6. Because the curve is above the line of equality, this ROC curve demonstrates that the random forest model improves the prediction precision over a random prediction model. The area under the ROC curve is 0.6826, which is larger than that obtained using a tree model (0.6817).

The optimal threshold calculated from the ROC curve is 0.367. The diagnostic properties of the classification rule using this threshold are shown in Table 4.2.

With the optimal threshold, the classification model predicts that 1785 individuals

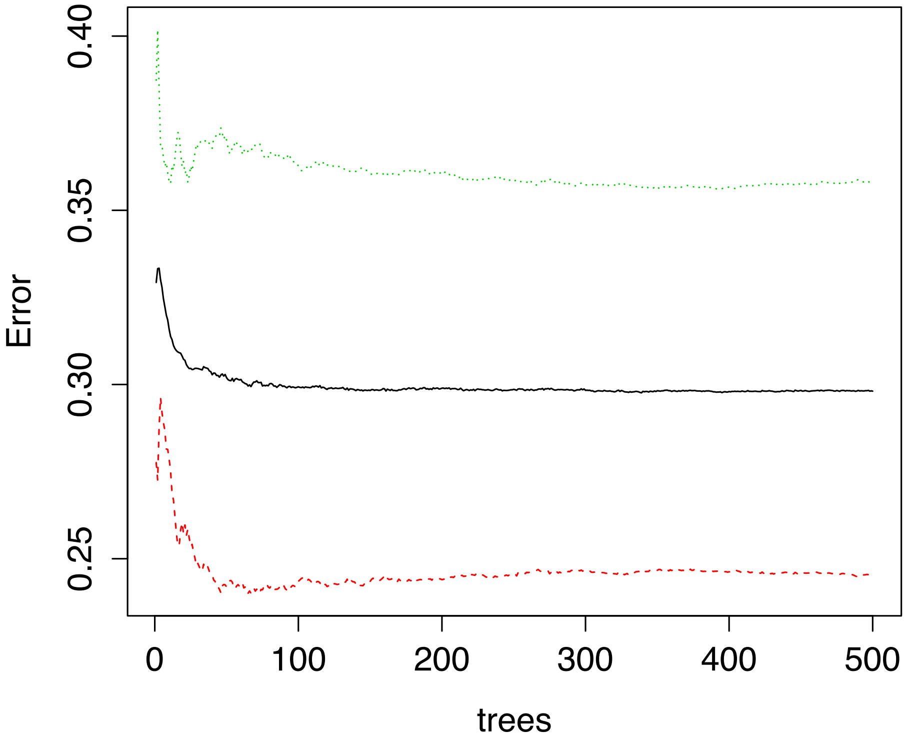

Figure 4.4: Random forest results for LGA. The OOB misclassification error (black line) is shown as a function of the number of classification trees. The number of predictors used for splitting at each node is $m = 4$. The green and red dashed lines represent the OOB error for $lga = 1$ and $lga = 0$ respectively.

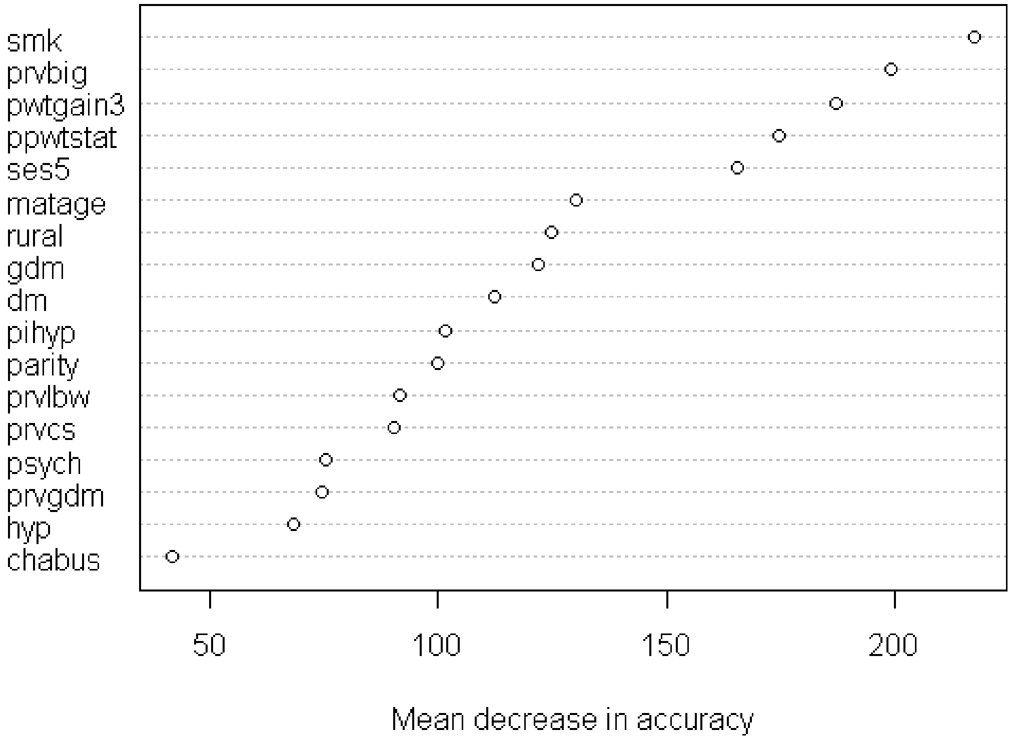

Figure 4.5: A variable importance plot for the LGA random forest. Variable importance is computed using the permutation approach.

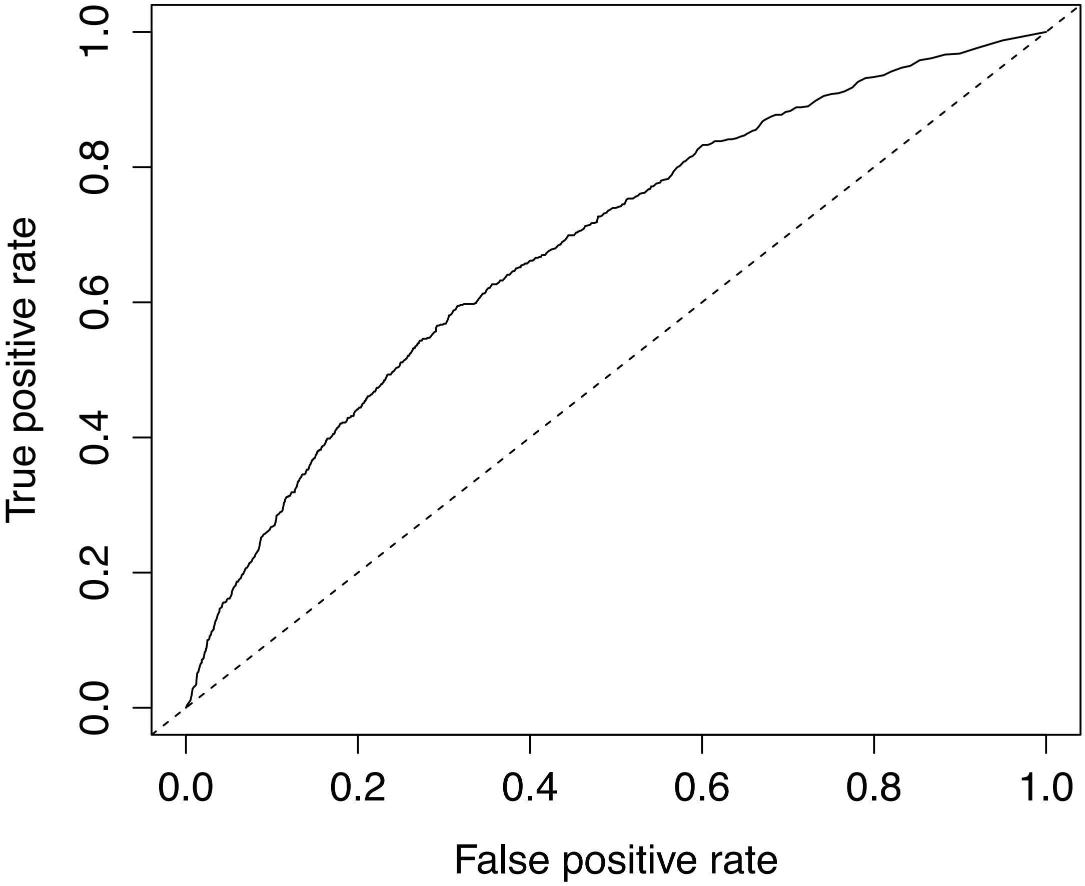

Figure 4.6: The ROC curve of LGA random forest.

	Disease+	Disease-	Total
Test+	429	1356	1785
Test-	289	2818	3107
Total	718	4174	4892

Point estimates and 95% CIs		
Apparent prevalence	0.36	(0.35, 0.38)
True prevalence	0.15	(0.14, 0.16)
Sensitivity	0.60	(0.56, 0.63)
Specificity	0.68	(0.66, 0.69)
Positive predictive value	0.24	(0.22, 0.26)
Negative predictive value	0.91	(0.90, 0.92)
Positive likelihood ratio	1.84	(1.71, 1.98)
Negative likelihood ratio	0.60	(0.54, 0.65)

Table 4.2: The confusion matrix and diagnostic summaries for LGA random forest with threshold 0.367.

will deliver LGA babies. Of the 718 mothers who deliver LGA babies, the classification model correctly predicts 429, or 60% (sensitivity). 1356 mothers who are not $lga = 1$ class are incorrectly classified. As a result, the overall error rate is 33.63%.

The positive likelihood ratio (1.84) and the negative likelihood ratio (0.60) indicate that a mother with a LGA baby is about 1.84 times more likely to have a positive test than a mother without a LGA baby, and the probability of having a negative test for mothers with LGA babies is 0.60 of that of those without LGA babies.

The random forest has overall better prediction performance than a tree model. The OOB misclassification error of a random forest is much lower than the cross validation error of a tree model. The area under the ROC curve of a random forest is larger than that of a tree model with higher sensitivity. However, the prediction performance of a random forest is somewhat limited. It may be improved in another tree ensemble method-boosting.

Chapter 5

Boosting Trees

Boosting is a general method for substantially improving the performance of a weak classifier or a weak regression model. It works by repeatedly applying a given weak learning algorithm to various distributions of the training data and then combing the classifiers or regression functions produced by the weak learners into a single classifier or regression model.

The first simple boosting algorithm was proposed by Schapire (1990), who proved that the performance of a weak classifier could be improved by training two other classifiers on a modified version of the training data and making the majority vote among these three classifiers. This was demonstrated by his "Strength of Weak Learnability" theorem. Based on the ideas presented by Schapire, Freund (1995) improved the performance of the simple boosting algorithm of Schapire by combining a large number of weak learners simultaneously. Freund and Shapire (1996) developed a new boosting algorithm, called AdaBoost. This commonly used method is more practical and easier to implement than the previous boosting algorithms.

The AdaBoost method was analyzed from a statistical view by Freidman et al. (2000). They used the exponential criterion and proved that the AdaBoost is actually an additive logistic model. Freidman (2001) developed gradient boosting and shrinkage for classification and regression.

In this chapter, the AdaBoost algorithm and its relationship with forward stagewise additive modeling are described. Boosting as applied to trees is discussed, and gradient boosting is presented. Some tuning parameters and the importance of variables are described. Gradient boosting with the AdaBoost loss criterion is applied to the NSAPD data set to predict SGA and LGA. The diagnostic properties of the fitted model are discussed.

5.1 Theory

5.1.1 AdaBoost and Forward Stagewise Additive Modeling

For a two- class problem, given data $(y_i, \mathbf{x}_i), i = 1, \ldots, n$, $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})$, the response variable y_i for each of observations is coded as $y_i \in \{-1, 1\}$. A classifier $G(\mathbf{x})$ for predictor variable $\mathbf{x}$ gives a prediction taking one of the two values $\{-1, 1\}$. For AdaBoost.M1 (also called Discrete AdaBoost), a weak classifier is sequentially applied on repeatedly weighted training data, thereby producing a sequence of weak classifiers. The final prediction is the weighted majority vote of the classifiers. Following Freidman et al. (2000), the details of this algorithm are shown as follows:

1. Start with weights as

$$\omega_i = 1/n, i = 1, 2, \ldots, n.$$

2. Repeat for $m = 1, 2, \ldots, M$.

(a) Fit the classifier $G_m(\mathbf{x})$ using weights ω_i, $i = 1, 2, \ldots, n$ on training data;

(b) Compute the weighted classification error rates

$$err_m = \frac{\sum_{i=1}^n \omega_i I(y_i \neq G_m(\mathbf{x}_i))}{\sum_{i=1}^n \omega_i}.$$

(c) Compute the weight

$$\alpha_m = log((1 - err_m)/err_m).$$

(d) Set weights

$$\omega_i' \leftarrow \omega_i \cdot exp[\alpha_m \cdot I(y_i \neq G_m(\mathbf{x}_i))], i = 1, 2, \ldots, n.$$

3. Output the classifier

$$G(\mathbf{x}) = sign[\sum_{m=1}^M \alpha_m G_m(\mathbf{x})].$$

For the AdaBoost.M1 algorithm, two sets of intervening weights are used. One set ω_i, $i = 1, 2, \ldots, n$ weights the observations. Another set α_m, $m = 1, 2, \ldots, M$ weights the contribution of each classifier. Their effects are discussed in the following.

1. For weights ω_i, $i = 1, 2, \ldots, n$, if the observations are correctly classified, then we have $I(y_i \neq G_m(x_i)) = 0$, and $\omega_i' \rightarrow \omega_i$; if the observations are misclassified, then we have $I(y_i \neq G_m(\mathbf{x}_i)) = 1$, and $\omega_i' \rightarrow \omega_i exp(\alpha_m)$. Since the weak classifiers are slightly better than random guessing, then we have $err_m < 0.5$ and $\alpha_m = log((1 - err_m)/err_m) > 0$. This indicates that more weight is applied to the misclassified observations and each successive classifier focuses more on those training observations which are misclassfied by the previous classifiers.

2. For weights α_m, $m = 1, \ldots, M$, according to $\alpha_m = log((1 - err_m)/err_m)$, smaller err_m leads to larger α_m. This implies that more weight is given to more accurate classifiers.

It has been shown that the AdaBoost method can dramatically improve the performance of weak classifiers. The mystery of this phenomenon was explained by Freidman et al. (2000). They showed that AdaBoost fits an additive model based on an exponential loss function, and the additive expansion produced by AdaBoost estimates the log-odds of the class probability.

Following Hastie et al. (2009), the additive models can be expressed by a set of basis function expansions

$$f(\mathbf{x}) = \sum_{m=1}^{M} \beta_m b(\mathbf{x}; \gamma_m),$$

where β_m, $m = 1, 2, \ldots M$ are the expansion coefficients, and $b(\mathbf{x}; \gamma_m)$ are basis function characterized by a set of parameters $\gamma_m, m = 1, 2, \ldots, M$. The basis functions here are chosen as the individual classifiers $G_m(\mathbf{x}) \in \{-1, 1\}$.

The additive models typically are fitted by minimizing a loss function averaged over the training data

$$\min_{\{\beta_m, \gamma_m\}_1^M} \sum_{i=1}^{N} L(y_i, \sum\nolimits_{m=1}^{M} \beta_m b(\mathbf{x}_i; \gamma_m)). \tag{5.1}$$

In many cases, the solution to (5.1) requires computationally intensive numerical optimization techniques. However, the computation can be substantially simplified by forward stagewise additive modeling, in which only one basis function and a corresponding coefficient are fitted at each iteration. Forward stagewise additive modeling starts from $f_0(\mathbf{x}) = 0$. For step m, $m = 1, \ldots M$, the optimal basis function and the corresponding coefficient are solved based on

$$(\beta_m, \gamma_m) = arg \min_{\beta,\gamma} \sum_{i=1}^{n} L(y_i, f_{m-1}(\mathbf{x}_i) + \beta b(\mathbf{x}_i; \gamma)),$$

and then the current expansion f_{m-1} is updated by

$$f_m(\mathbf{x}) = f_{m-1}(\mathbf{x}) + \beta_m b(\mathbf{x}; \gamma_m).$$

This process is repeated without modifying the previously added terms.

For AdaBoost.M1, the basis function are the individual classifiers $G_m(\mathbf{x}) \in \{-1, 1\}$. The exponential loss function is given by

$$L(y, f(\mathbf{x})) = exp(-yf(\mathbf{x})) \tag{5.2}$$

where $f(x)$ is a real number. In this case, it is a weighted sum of classifiers. When $y = 1$, $L = exp(-f)$, so if $f(x)$ is large and positive, then L is small, and the classifier given by $sign(f)$ would be 1. But if $f(x)$ is large and negative, L is large, and the classifier would give -1 which would be an error.

The optimal classifier G_m and the coefficient β_m are solved by

$$(\beta_m, G_m) = arg \min_{\beta,G} \sum_{i=1}^{n} exp[-y_i(f_{m-1}(\mathbf{x}_i) + \beta G(\mathbf{x}_i))].$$

This can be expressed as

$$(\beta_m, G_m) = arg \min_{\beta,G} \sum_{i=1}^{n} \omega_i^{(m)} exp[-\beta y_i G(\mathbf{x}_i)], \tag{5.3}$$

where $\omega_i^{(m)} = exp[-y_i f_{m-1}(\mathbf{x}_i)]$. Each $\omega_i^{(m)}$ depends on neither β nor $G(\mathbf{x})$, and therefore, it can be considered as weights applied to each observation. The observation weights change with each iteration m as they are determined by $f_{m-1}(\mathbf{x}_i)$.

Equation (5.3) can also be expressed as

$$\begin{aligned}(\beta_m, G_m) =& arg\min_{\beta,G}[e^{-\beta} \cdot \sum_{y_i=G(\mathbf{x}_i)} \omega_i^{(m)} + e^{\beta} \cdot \sum_{y_i \neq G(\mathbf{x}_i)} \omega_i^{(m)}] \\ =& arg\min_{\beta,G}[(e^{\beta} - e^{-\beta}) \cdot \sum_{i=1}^{n} \omega_i^{(m)} I(y_i \neq G(\mathbf{x}_i)) + e^{-\beta} \cdot \sum_{i=1}^{n} \omega_i^{(m)}].\end{aligned} \tag{5.4}$$

For $\beta > 0$, the solution to (5.4) for $G_m(\mathbf{x})$ is given by

$$G_m = arg\min_{G} \sum_{i=1}^{n} \omega_i^{(m)} I(y_i \neq G(\mathbf{x}_i)), \tag{5.5}$$

which is the classifier minimizing the weighted error rate in predicting y.

The minimized weighted error rate, denoted by err_m is

$$err_m = \frac{\sum_{i=1}^{N} \omega_i^{(m)} I(y_i \neq G_m(\mathbf{x}_i))}{\sum_{i=1}^{n} \omega_i^{(m)}}.$$

Substituting G_m into (5.3) and solving for β, we obtain

$$\beta_m = \frac{1}{2} log\left[\frac{1 - err_m}{err_m}\right],$$

which is the same as the expression of err_m in the AdaBoost.M1 algorithm.

The expansion at iteration m is then updated by

$$f_m(\mathbf{x}) = f_{m-1}(\mathbf{x}) + \beta_m G_m(\mathbf{x})$$

and the weights for the next iteration $m + 1$ updated correspondingly

$$\omega_i^{(m+1)} = \omega_i^{(m)} \cdot e^{-\beta_m y_i G_m(\mathbf{x}_i)}. \tag{5.6}$$

Using the fact $-y_i G_m(\mathbf{x}_i) = 2 \cdot I(y_i \neq G_m(\mathbf{x}_i)) - 1$, and ignoring the common factor $e^{-\beta_m}$ for all weights, the equation (5.6) becomes

$$\omega_i^{(m+1)} = \omega_i^{(m)} \cdot e^{-\alpha_m I(y_i \neq G_m(\mathbf{x}_i))},$$

where $\alpha_m = 2\beta_m$. This is exactly the same as the expression for ω_i in the AdaBoost.M1 algorithm. Thus, we can conclude that AdaBoost.M1 is equivalent to the forward stagewise additive model which minimizes the exponential loss criterion.

The exponential loss criterion (5.2) proposed by Schapire and Singer (1998) has special statistical properties (Friedman et al. 2000).

Consider minimizing the criterion $E(e^{-yf(\mathbf{x})})$ for estimation of $f(\mathbf{x})$,

$$f^*(\mathbf{x}) = arg \min_{f(\mathbf{x})} E_{y|\mathbf{x}}(e^{yf(\mathbf{x})}) = \frac{1}{2} log \frac{P(y=1|\mathbf{x})}{P(y=-1|\mathbf{x})},$$

where $E(e^{-yf(\mathbf{x})})$ represents the expectation of $e^{-yf(\mathbf{x})}$ and it is sufficient to be minimized conditional on $\mathbf{x}$.

The expectation is

$$E[e^{-yf(\mathbf{x})}|\mathbf{x}] = P(y=1|\mathbf{x})e^{-f(\mathbf{x})} + P(y=-1|\mathbf{x})e^{f(\mathbf{x})}$$

and its derivative is

$$\frac{dE[e^{-yf(\mathbf{x})}|\mathbf{x}]}{df(\mathbf{x})} = -P(y=1|\mathbf{x})e^{-f(\mathbf{x})} + P(y=-1|\mathbf{x})e^{f(\mathbf{x})}.$$

Setting this derivative to zero gives the result

$$f^*(\mathbf{x}) = \frac{1}{2} log \frac{P(y=1|\mathbf{x})}{P(y=-1|\mathbf{x})},$$

which is one-half the log-odds of $P(y=1|\mathbf{x})$. Hence, we have

$$P(y=1|\mathbf{x}) = \frac{e^{f(\mathbf{x})}}{e^{-f(\mathbf{x})} + e^{f(\mathbf{x})}}$$

and

$$P(y=-1|\mathbf{x}) = \frac{e^{-f(\mathbf{x})}}{e^{-f(\mathbf{x})} + e^{f(\mathbf{x})}}.$$

We can further conclude that the AdaBoost.M1 is actually an additive logistic regression model.

Another useful loss criterion is the binomial negative log-likelihood or deviance. It can be shown that this has the same population minimizer as the exponential loss criterion (Frieman et al., 2000).

Let $y' = (y+1)/2$, taking values 0,1, and parameterize the binomial probabilities by

$$p(\mathbf{x}) = P(y=1|\mathbf{x}) = \frac{e^{f(\mathbf{x})}}{e^{-f(\mathbf{x})} + e^{f(\mathbf{x})}}. \tag{5.7}$$

The binomial log-likelihood is

$$l(y, f(\mathbf{x})) = y' logp((\mathbf{x})) + (1-y')log(1-p(\mathbf{x})),$$

or equivalently the deviance is

$$-l(y, f(\mathbf{x})) = log(1 + e^{-2yf(\mathbf{x})}).$$

Since the population minimizers of the binomial negative log-likelihood or deviance is at the true probabilities $p(\mathbf{x})$ given by (5.7), the population minimizers of the deviance $E_{y|\mathbf{x}}(-l(y, f(\mathbf{x}))$ and $E_{y|\mathbf{x}}(e^{-yf(\mathbf{x})})$ are the same. However, for a finite data set, they are not the same. The deviance is far more robust than exponential loss in situations where misspecification of the class label appears in the training data (Hastie et al., 2009).

For regression problems, the typical loss function in an additive model is the squared-error loss function. It is of the form

$$\begin{aligned} L(y_i, f_{m-1}(\mathbf{x}_i) + \beta b(\mathbf{x}_i;\gamma)) =& (y_i - f_{m-1}(\mathbf{x}_i) - \beta b(\mathbf{x}_i;\gamma))^2 \\ =& (r_{im} - \beta b(\mathbf{x}_i;\gamma))^2, \end{aligned} \tag{5.8}$$

where $r_{im} = y_i - f_{m-1}(\mathbf{x}_i)$ is the residual of the current model on the i th observation. Thus the term $\beta_m b(\mathbf{x};\gamma)$ added to the expansion at iteration m is actually the best fit to the current residuals.

The squared-error loss for the finite sample, however, is far less robust for long-tailed error distributions and especially for outliers as it penalizes heavily observations with large absolute residuals $|y_i - f(\mathbf{x}_i)|$ during the fitting process. Other losses such as absolute loss are more robust in these situations. The Huber loss criterion

(Huber,1964) is better than both loss for error distribution. It combines the good properties of squared-error loss near zero and absolute error loss when $|y - f|$ is large (Hastie et al., 2009).

On the other hand, using a more robust loss function for a classification or regression additive model does not create as simple a modular algorithm as exponential or squared-error loss. This problem has been solved using a gradient boosting algorithm that is based on any differentiable loss criterion.

5.1.2 Boosted Tree Model

The boosted tree model can be treated as a special case for the additive model when the basis functions are chosen as a single classification or regression tree. Following Hastie et al. (2009), a single tree model can be presented by disjoint regions R_j, $j = 1, 2, \ldots, J$, with a constant prediction γ_j assigned to each terminal node. Thus a tree can be formally expressed as

$$T(\mathbf{x}; \Theta) = \sum_{j=1}^{J} \gamma_j I(\mathbf{x} \in R_j),$$

with parameters $\Theta = \{R_j, \gamma_j\}_1^J$ characterizing a tree in terms of split variables, cutpoints at each node, and terminal-node values.

The boosted tree model can therefore be presented by

$$f_M(\mathbf{x}) = \sum_{m=1}^{M} T(\mathbf{x}; \Theta_m), \tag{5.9}$$

which is the sum of single trees created in a forward stagewise manner. At each iteration m, $m = 1, 2, \ldots, M$, the parameters $\Theta_m = \{R_j, \gamma_j\}_1^J$ are estimated by minimizing a loss function based on the current expansion $f_{m-1}(\mathbf{x})$

$$\hat{\Theta}_m = arg \min_{\Theta_m} \sum_{i=1}^{n} L(y_i, f_{m-1}(\mathbf{x}_i) + T(\mathbf{x}_i; \Theta)). \tag{5.10}$$

Given R_{jm}, γ_{jm} can be solved from

$$\hat{\gamma}_{jm} = arg \min_{\gamma_{jm}} \sum_{\mathbf{x}_i \in R_{jm}} L(y_i, f_{m-1}(\mathbf{x}_i) + \gamma), \tag{5.11}$$

and then the expansion is updated by $f_m(\mathbf{x}) = f_{m-1}(\mathbf{x}) + T(\mathbf{x}_i; \Theta_m)$.

For a two-class boosted classification tree, using exponential loss, we obtain a similar result using $T(\mathbf{x}_i; \theta_m)$ instead of $G(\mathbf{x}_i)$ in (5.3),

$$\hat{\Theta}_m = arg \min_{\Theta_m} \sum_{i=1}^{n} \omega_i^{(m)} exp[-y_i T(\mathbf{x}_i; \Theta_m)],$$

with weights $\omega_i^{(m)} = e^{-y_i f_{m-1}(\mathbf{x}_i)}$. This indicates that a new tree is produced based on a greedy recursive-partitioning algorithm using this weighted exponential loss function as a splitting criterion.

For a scaled classification tree, which is a classification tree $T(\mathbf{x}; \Theta_m)$ with the restriction $\gamma_{jm} \in \{-1, 1\}$, we obtain

$$\hat{\Theta}_m = arg \min_{\Theta_m} \sum_{i=1}^{n} \omega_i^{(m)} I(y_i \neq T(\mathbf{x}_i; \Theta_m))$$

with weights $\omega_i^{(m)} = e^{-y_i f_{m-1}(\mathbf{x}_i)}$.

Given the R_{jm}, the solution to (5.11) can be derived as

$$\hat{\gamma}_{jm} = \frac{1}{2} log \left(\frac{\sum_{\mathbf{x}_i \in R_{jm}} \omega_i^{(m)} I(y_i = 1)}{\sum_{\mathbf{x}_i \in R_{jm}} \omega_i^{(m)} I(y_i = -1)} \right),$$

which is a weighted log-odds in R_{jm}, $j = 1, 2, \ldots, J$.

For a boosted regression tree using squared-error loss, the solution to (5.10) is

$$\begin{aligned} \hat{\Theta}_m =& arg \min_{\Theta_m} \sum_{i=1}^{n} (y_i - f_{m-1}(\mathbf{x}_i) - T(\mathbf{x}_i; \Theta_m))^2 \\ =& arg \min_{\Theta_m} \sum_{i=1}^{n} (r_{im} - T(\mathbf{x}_i; \Theta_m))^2. \end{aligned} \tag{5.12}$$

Given R_{jm}, γ_{jm} can be estimated using (5.11),

$$\hat{\gamma}_{jm} = mean[y_i - f_{m-1}(\mathbf{x}_i) | \mathbf{x}_i \in R_{jm}]. \tag{5.13}$$

As mentioned in the previous section, for more general robust loss functions, an algorithm for boosting trees is not as simple and fast as an exponential loss function for classification and a squared-error loss function for regression. However, gradient boosting can fix this problem.

5.1.3 Gradient Boosting Trees

A gradient boosting method was first introduced by Friedman (2001). It made the connection between the stagewise additive expansion and steepest-descent minimization. It is a general gradient descent "boosting" method developed for the additive expansion based on any differentiable loss criterion.

Following Hastie et al (2009), the general loss function in using $f(\mathbf{x})$ to predict y on the training data can be written by

$$L(f) = \sum_{i=1}^{n} L(y_i, f(\mathbf{x}_i)).$$

For numerical optimization in function space, the constraint for $f(\mathbf{x})$ to be a sum of trees (5.9) is ignored. Consider $\mathbf{f}$ evaluated at each point $\mathbf{x}$ to be a "parameter" and search to minimize

$$\hat{\mathbf{f}} = arg \min_{\mathbf{f}} L(\mathbf{f}),$$

where the $\mathbf{f} \in \mathbb{R}^n$ are the values of the approximating function $f(\mathbf{x}_i)$ at each of the n data points $\mathbf{x}_i$:

$$\mathbf{f} = \{f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_n)\}^T.$$

Following the numerical optimization procedures, we take the solution to be

$$\mathbf{f}_M = \sum_{i=0}^{M} \mathbf{h}_m,$$

where $\mathbf{f}_0 = \mathbf{h}_0$ is an initial guess, and $\mathbf{h}_m$ are the incremental vectors ("step" or "boots") defined by various optimization methods.

One of the simplest of the commonly used numerical optimization methods is steepest descent or gradient descent in which the steps are taken proportional to the negative of the gradient of the function at the current point to achieve a local minimum of a function. Thus, for steepest descent,

$$\mathbf{h}_m = -\rho_m \mathbf{g}_m$$

where ρ_m is step length and $\mathbf{g}_m$ is the gradient of $L(\mathbf{f})$ evaluated at $\mathbf{f} = \mathbf{f}_{m-1}$. The components of the gradient $\mathbf{g}_m$ are

$$g_{im} = [\frac{\partial L(y_i, f(\mathbf{x}_i))}{\partial f(\mathbf{x}_i)}]_{f=f_{m-1}}, \tag{5.14}$$

and ρ_m is given by line search

$$\rho_m = arg \min_{\rho} L(\mathbf{f}_{m-1} - \rho \mathbf{g}_m).$$

The current incremental function is updated

$$\mathbf{f}_m = \mathbf{f}_{m-1} - \rho_m \mathbf{g}_m.$$

The gradient descent is a very greedy strategy because $-\mathbf{g}_m$ is the local direction in $\mathbb{R}^n$ in which the loss function $L(\mathbf{f})$ has the most decrease at $\mathbf{f}_{m-1}$. However, the gradient given by (5.14) is only defined at the training data point $\mathbf{x}_i$, while the $\mathbf{f}_M$ must be generated for new data points for prediction.

In order to solve this problem, the gradient descent minimization is combined with the forward stagewise boosting tree model. At each iteration m, $m = 1, 2, \ldots, n$, a tree $T(\mathbf{x}; \Theta_m)$ is produced by fitting the negative gradient values. Using least squared error as the fitting criterion, we obtain

$$\tilde{\Theta}_m = arg \min_{\theta} \sum_{i=1}^{n} (-g_{im} - T(\mathbf{x}_i; \Theta))^2. \tag{5.15}$$

After generating the tree (5.15), the constants in each node are estimated by (5.11). Using the gradient boosting method, boosting trees can be created with any differentiable loss function. As Friedman (2001) presented, gradient boosting of regression trees produces a competitive, highly robust and interpretable procedure for both classification and regression.

5.1.4 Tuning Parameters and Relative Importance of Predictor Variables

Boosting has three tuning parameters: the size of each of the tree J_m, $m = 1, 2, \ldots, M$, the number of boosting iterations or trees M, and the shrinkage parameter λ.

The size of each tree controls the complexity of the boosted ensemble. In practice, all the trees in the ensemble are restricted to be the same size J in order to prevent too large trees being produced in the tree boosting procedure, which can result in poor performance and intensive computation. The size of tree J is a meta-parameter that can be tuned to minimize the prediction error.

The size of tree J or the number of splits $d = J - 1$ also controls the interaction order of the boosted model. The number of splits d is defined as the interaction depth since d splits involves at most d variables. In many cases, the low-order interactions are dominant, therefore d tends to be low. $d = 1$ usually performs well, in which case each tree in the ensemble is a stump. This corresponds to an additive model containing only one variable in each term.

The number of boosting iterations or trees M is another meta-parameter of gradient boosting. The prediction risk is reduced at each iteration and can be arbitrarily small when M is large enough. However, large M can cause overfitting and leads to high prediction risk for test data. Therefore, the optimal value of M needs to be adjusted. In practice, the prediction risk as a function of M can be monitored by cross validation in the entire gradient boosting process. The estimated optimal value of M is determined when the minimum risk is achieved.

A shrinkage technique can substantially improve the performance in a boosting model (Friedman, 2001). It is implemented in the boosting procedure by scaling the contribution of each tree by a factor λ $(0 < \lambda < 1)$ when it is added to the current approximation,

$$f_m(\mathbf{x}) = f_{m-1}(\mathbf{x}) + \lambda \sum\nolimits_{j=1}^{J_m} \gamma_{jm} I(\mathbf{x} \in Rjm).$$

The shrinkage λ controls the rate at which the boosting learns, and allows more and different trees to get involved in the boosting process. Friedman (2001) revealed that small values of λ can lead to small test error and large values of M are required. Therefore, both λ and M control prediction risk on the training data and there is a trade-off between them. Usually shrinkage is set to be very small and the optimal value of M is then chosen. The typical values of λ are 0.01 and 0.001.

Relative importance of predictor variables, also called relative influence is the most useful description of the boosting tree model. An extension of this feature for boosted estimates was developed by Friedman (2001). For a single tree, Breiman et al. (1984) proposed a measure of relevance for each predictor variable as

$$I_l^2(T) = \sum_{t=1}^{J-1} \hat{i}_t^2 \; I(v(t) = l) \tag{5.16}$$

where the summation is over the internal nodes t of the J-terminal node tree T, $v(t)$ is the splitting variable associated with node t, and $\hat{i}_t^2$ is the corresponding empirical improvement in squared error for regression, or in impurity index for classification, as a result of the split.

For an additive tree expansion, (5.16) can be generalized by its average over all trees

$$I_l^2 = \frac{1}{M} \sum_{m=1}^{M} I_l^2(T_m).$$

The importance measure is more stable than for a single tree due to the averaging. It also provides the summarized and quantitative information of the boosting tree and discloses the relevance of a predictor to the response variable.

5.2 Results

5.2.1 SGA

We construct gradient boosting classification trees on the training data for SGA using the **gbm** function in the gbm package in R. The criterion for boosting trees in each iteration is based on the AdaBoost exponential loss function (AdaBoost exponential bound).

The three tuning parameters are chosen as follows: the size of each classification tree is $J_m = 2$, $m = 1, 2, \ldots, M$, or the interaction depth is $d = 1$ indicating each tree in the iterations is a stump, with only one variable included in each tree; the shrinkage is the lower typical value $\lambda = 0.001$, and the number of boosting iterations or trees is $M = 20,000$. These parameters maximized the AUC for the training data.

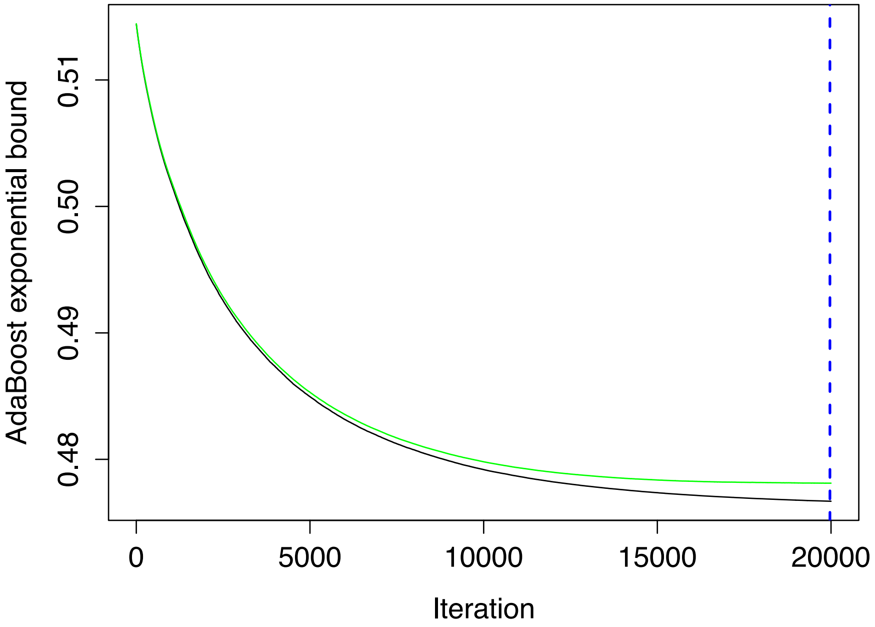

Figure 5.1: AdaBoost exponential loss function as a function of number of iterations for SGA. The black line plots the estimate of the loss function for each boosting iteration evaluated on the training data as a function of number of iterations for SGA. The green line plots the 5-folds cross validation estimate of the loss function for each boosting iteration as a function of number of iterations for SGA.

Figure 5.1 shows the AdaBoost exponential loss function progression on 20000 boosted trees using 5-fold cross validation. The black line demonstrates the estimate of the loss function for each boosting iteration evaluated on the training data as a function of number of iterations. The green line demonstrates the cross validation estimate of the loss function for each boosting iteration as a function of number of iterations. It appears that the cross validation estimate of the loss function decreases monotonically with increasing M and stabilizes at 19969 trees. Therefore, the optimal number of boosting iterations for predicting SGA in this parameter setting is 19969.

Table 5.1 presents the rankings of the relative variable importance for each of predictor variables, and also displays in Figure 5.2.

smk and *pwtgain*3 are the most relevant predictors. *parity*, *ppwtstat*, *prvbig* , *prvlbw*, *prvlbm*, and *pihyp* have roughly one fourth of the relevance of *smk*, whereas others are less influential.

Variable	Relative Influence
smk	32.4317
*pwtgain*3	24.4285
parity	8.9585
ppwtstat	8.2781
prvbig	7.007
prvlbw	6.0822
pihyp	5.2781
hyp	2.4738
*ses*5	1.7909
matage	1.7340
prvcs	0.5314
chabus	0.4149
gdm	0.2993
dm	0.1345
psych	0.0733
rural	0.0431
prvgdm	0.0408

Table 5.1: Relative influence of the predictors for SGA.

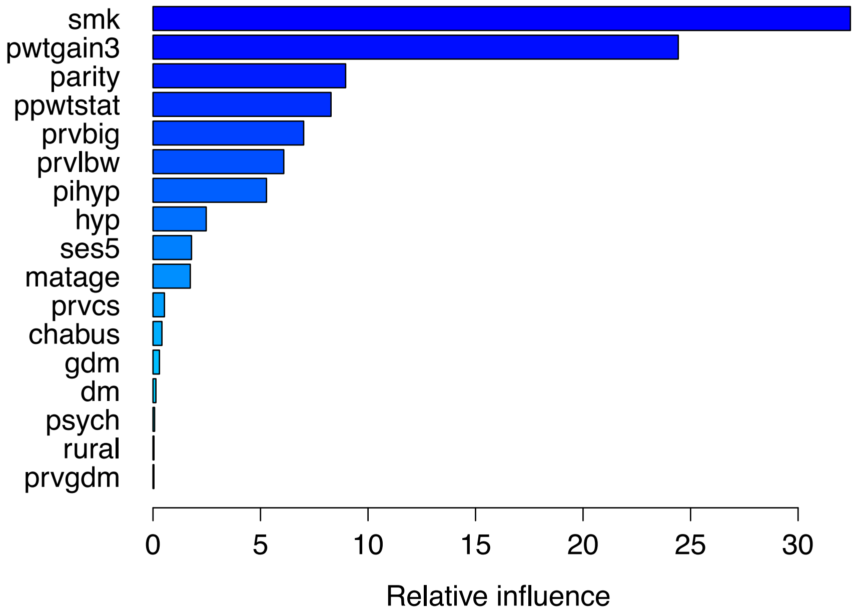

Figure 5.2: Variable relative influence plot for SGA boosted trees.

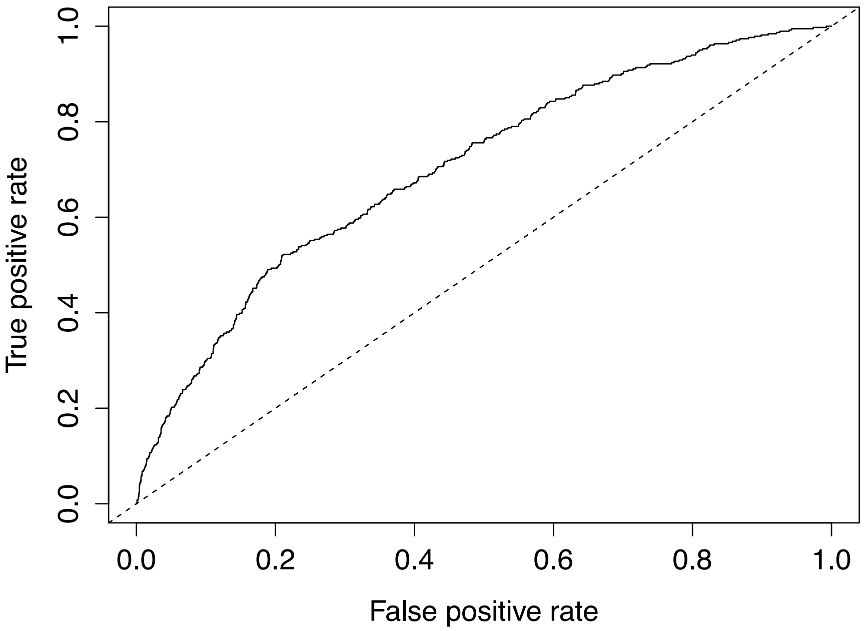

Figure 5.3: The ROC curve of SGA boosted trees.

The ROC curve of the boosted tree model for SGA on the test data is plotted in Figure 5.3. Because the curve is above the line of equality, this ROC curve demonstrates that the random forest model improves the prediction precision over a random prediction model. The area under the ROC curve is 0.7025, which is larger than that obtained using a random forest model (0.6752).

The optimal threshold calculated from the ROC curve is 0.065. The diagnostic properties of the classification rule using this threshold are shown in Table 5.2.

With the optimal threshold, the classification model predicts that 1923 individuals will deliver SGA babies. Of the 381 individuals who actually deliver SGA babies, the classification model correctly predicts 251, or 66% (sensitivity). 1672 individuals who are not $sga = 1$ class are incorrectly classified. As a result, the overall error rate has increased to 36.84%.

The positive likelihood ratio (1.78) and the negative likelihood ratio (0.54) indicate that a mother with a SGA baby is about 1.78 times more likely to have a positive test than a mother without a SGA baby, and the probability of having a negative test for mothers with SGA babies is 0.54 of that of those without SGA babies.

	Disease+	Disease-	Total
Test+	251	1672	1923
Test-	130	2839	2969
Total	381	4511	4892

Point estimates and 95% CIs		
Apparent prevalence	0.39	(0.38, 0.41)
True prevalence	0.08	(0.07, 0.09)
Sensitivity	0.66	(0.61, 0.71)
Specificity	0.63	(0.62, 0.64)
Positive predictive value	0.13	(0.12, 0.15)
Negative predictive value	0.96	(0.95, 0.96)
Positive likelihood ratio	1.78	(1.64, 1.93)
Negative likelihood ratio	0.54	(0.47, 0.62)

Table 5.2: The confusion matrix and diagnostic summaries for SGA boosted trees with threshold 0.065.

5.2.2 LGA

We construct gradient boosting classification trees for LGA using the **gbm** function in the *gbm* package in R on training data. The criterion for boosting trees in each iteration is based on the AdaBoost exponential loss function (AdaBoost exponential bound).

The three tuning parameters are chosen as follows: the size of each classification tree is $J_m = 3$, $m = 1, 2, \ldots, M$, or the interaction depth is $d = 2$ indicating that each tree in the iterations contains two splitting variables with interaction between them; the shrinkage is the upper typical value $\lambda = 0.01$, and the number of boosting iterations or trees $M = 3000$. These parameters maximized the AUC for the training data.

Figure 5.4 shows the AdaBoost exponential loss function progression on 3000 boosted trees using 5-fold cross validation. The black line demonstrates the estimate of the loss function for each boosting iteration evaluated on the training data as a function of number of iterations for LGA. The green line demonstrates the cross validation estimate of the loss function for each boosting iteration as a function of number of iterations for LGA. The cross validation estimate of the loss function decreases monotonically with increasing M and stabilizes at 1461 trees. Therefore,

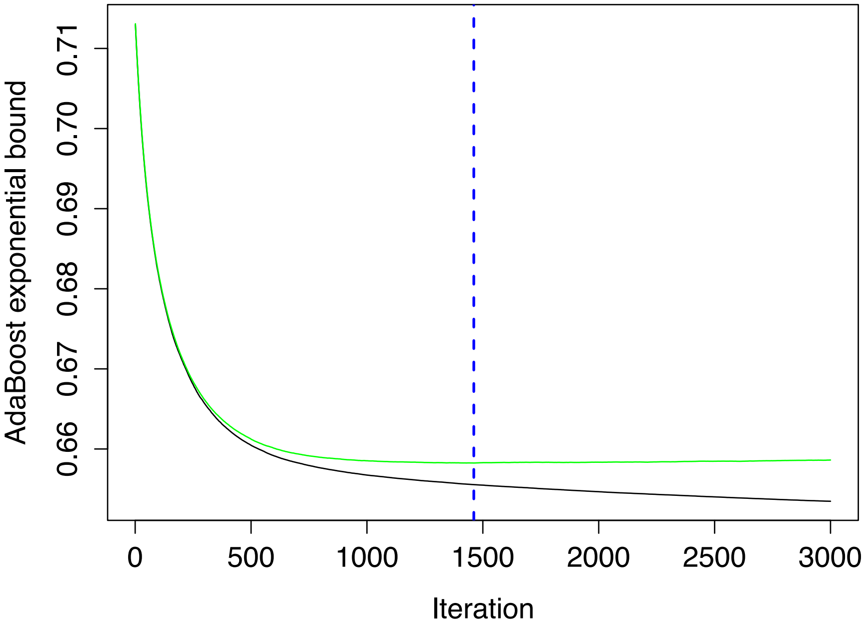

Figure 5.4: AdaBoost exponential loss function as a function of number of iterations for LGA. The black line plots the estimate of the loss function for each boosting iteration evaluated on the training data as a function of number of iterations for LGA. The green line plots the 5-folds cross validation estimate of the loss function for each boosting iteration as a function of number of iterations for LGA.

the optimal number of boosting iterations for predicting LGA in this parameter setting is 1461.

Table 5.3 presents the relative variable importance for each of predictor variables, which is also also displayed in Figure 5.5.

*prvbig*is the most relevant predictor. *pwtgain*3,*ppwtstat*, and *smk* have roughly one third of the relevance of *smk*, whereas others are less influential.

The ROC curve of the boosted tree model for LGA on the testing data is plotted in Figure 5.6. Because the curve is above the line of equality, this ROC curve demonstrates that the gradient boosted tree model improves the prediction precision over a random prediction model. The area under the ROC curve is 0.7107, which is larger than that obtained using a random forest model (0.6826).

The optimal threshold calculated from the ROC curve is 0.155. The diagnostic properties of the classification rule using this threshold is shown in Table 5.4.

Variable	Relative Influence
prvbig	42.3925
*pwtgain*3	13.9736
ppwtstat	13.3144
smk	8.5578
matage	4.8941
dm	4.3784
parity	3.1725
gdm	2.7451
*ses*5	2.2117
prvlbw	1.1689
rural	0.6864
pihyp	0.6760
prvgdm	0.5207
prvcs	0.4884
hyp	0.4373
psych	0.4373
chabus	0.1175

Table 5.3: Relative importance of the predictors for LGA.

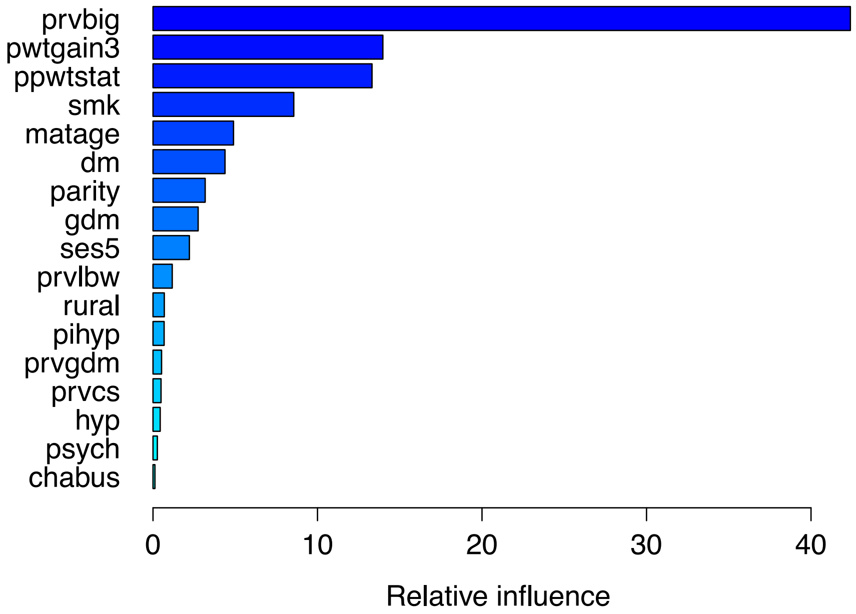

Figure 5.5: Variable relative influence plot for LGA boosted trees.

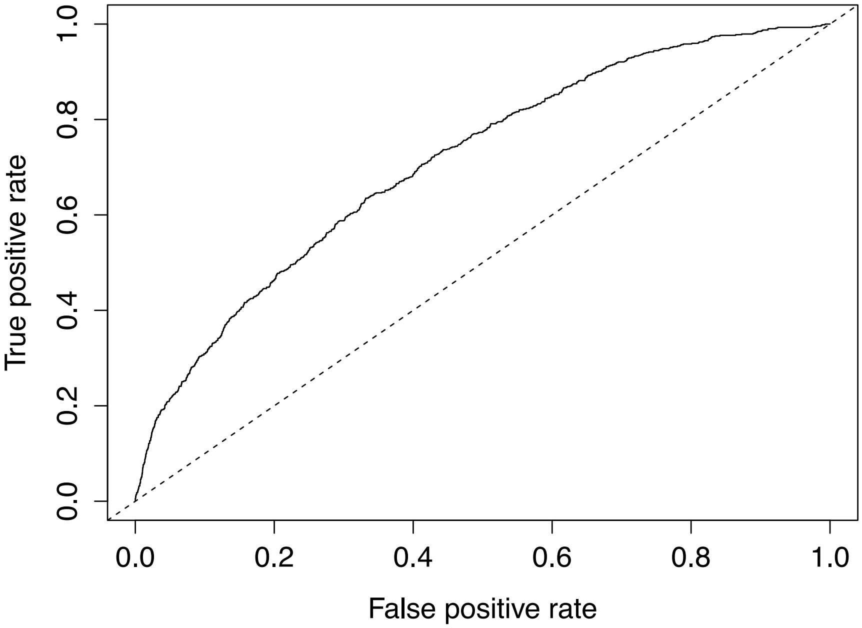

Figure 5.6: The ROC curve of LGA boosted trees.

	Disease+	Disease-	Total
Test+	455	1383	1838
Test-	263	2791	3054
Total	718	4174	4892

Point estimates and 95% CIs		
Apparent prevalence	0.38	(0.36, 0.39)
Ture prevalence	0.15	(0.14, 0.16)
Sensitivity	0.63	(0.60, 0.67)
Specificity	0.67	(0.65, 0.68)
Positive predictive value	0.25	(0.23, 0.27)
Negative predictive value	0.91	(0.90, 0.92)
Positive likelihood ratio	1.91	(1.78, 2.05)
Negative likelihood ratio	0.55	(0.50, 0.60)

Table 5.4: The confusion matrix and diagnostic summaries for LGA boosted trees with threshold 0.155.

With the optimal threshold, the classification model predicts that 1838 individuals will deliver SGA babies. Of the 718 individuals who deliver LGA babies, the classification model correctly predicts 455, or 63% (sensitivity). 1383 individuals who are not $lga = 1$ are incorrectly classified. As a result, the overall error rate has increased to 33.65%.

The positive likelihood ratio (1.91) and the negative likelihood ratio (0.55) indicate that a mother with a LGA baby is about 1.91 times more likely to have a positive test than a mother without a LGA baby, and the probability of having a negative test for mothers with LGA babies is 0.55 of that of those without LGA babies.

Chapter 6

Summary and Discussion

Several statistical prediction methods have been reviewed in this study including logistic regression, classification and regression trees, random forest and boosted tree models. Their application to predicting fetal growth abnormalities such as SGA and LGA using the NSAPD dataset are presented.

Logistic regression is the most commonly used method for predicting a binary variable in health science research. The model itself can be systemically constructed and easily interpreted based on the concept of odds ratio. However, for large amounts of data, both in terms of a large number of observations and a large number of predictor variables, especially in messy cases, with a mixture of different types of predictors, with long tailed and highly skewed distributions of numerical predictors, and with a number of irrelevant prediction variables, the data preprocessing and model fitting is complex and time consuming.

The decision tree method can deal with those problems and construct a predictive model and make predictions very quickly. From the tree fitted procedure, the mixtures of different types of predictors, missing data are accommodated by the algorithm. The internal splitting variable selection as a part of this procedure prevents inclusion of many irrelevant predictors. Furthermore the decision tree model can be graphically presented and easily explained. These attractive properties have made it become the most popular predictive model.

However the downside of the decision tree model is its high variance and lower predictive accuracy. Usually a small change in the data can cause very different splits in the tree construction. These problems are dramatically alleviated by the tree ensemble-based methods, random forest and boosting tree models.

Both random forest and boosted tree model generate a diverse ensemble of trees by repeatedly manipulating the training data, and combine these computed trees into a single predictive model. However, they are fundamentally different. Randomization is an essential feature for the random forest model. Each tree in the ensemble is constructed using a bootstrapped sample from the training data, which is randomly selected with replacement form original training data. In addition, at each node, a fraction of predictors are randomly chosen from all predictors in order to decorrelated each tree, which reduces the variance. The final model is formed by giving equal weight to each random forest tree. On the other hand, Adaboost is a deterministic algorithm. Each observation is iteratively assigned a weight based on how it was classified by the previous classifier. More weight is applied to the observations miss-classified by the previous tree, and this information is learned by the next tree. The final model is a weighted majority vote of the sequence of classifiers with more weight assigned to more accurate trees. In this way, the boosted tree model can substantially improve the predictive accuracy of individual trees and also outperform random forest in predictive accuracy.

Moreover, both random forest and boosted tree model can substantially reduce the variance of a single tree model. However, boosting trees can also reduce the bias for each individual tree (Freund, 1996), while the bias of random forest is the same as the bias of any individual tree.

All these methods are applied to the NSAPD data for predicting SGA and LGA. The models were fitted to the training data set, which was randomly selected at the beginning of the study. Their prediction performance is assessed using the test data, which is the remainder of the full data set. Some diagnostic properties for predicting SGA by the four classifiers are shown in Table 6.1.

Prediction model	Threshold	MCE	Sensitivity	Specificity	AUC
Logistic regression	0.065	0.3659	0.65	0.63	0.6990
Classification tree	0.5	0.2796	0.52	0.74	0.6603
Random forest	0.276	0.3265	0.60	0.68	0.6752
Boosting tree	0.065	0.3684	0.66	0.63	0.7025

Table 6.1: Diagnostic properties of four classifiers for predicting SGA. The threshold, the misclassification error rate (MCE), sensitivity, specificity and area under the ROC curve (AUC) are presented.

Comparing with logistic regression for predicting SGA, the boosted tree model has greater accuracy (0.7024) when assessing performance using AUC. Random forest and tree models have less accuracy than logistic regression (0.6990). The tree model has the least accuracy (0.6603). In addition, the boosted tree model has greater sensitivity (0.66) than that (0.65) of logistic regression. Random forest and tree models have less sensitivity than that of logistic regression. The tree model has the least sensitivity (0.52).

Ranking	Logistic regression	Classification tree	Random forest	Boosting tree
1	smk	smk	smk	smk
2	pwtgain3	pwtgain3	pwtgain3	pwtgain3
3	pihyp	parity	parity	parity
4	parity	prvbig	ses5	ppwtstat
5	prvlbw	ppwtstat	ppwtstat	prvbig
6	prvbig	prvlbw	rural	prvlbw
7	ppwtstat	matage	prvlbw	pihyp
8	hyp	ses5	matage	hyp
9	ses5		prvbig	ses5
10	prvcs		gdm	matage
11	gdm		pihyp	prvcs
12	matage		hyp	chabus
13	dm		psych	gdm
14			prvcs	dm
15			prvgdm	psych
16			dm	rural
17			chabus	prvgdm

Table 6.2: Variable rankings in four methods for predicting SGA. Variables are ranked from high relevance to low relevance based on absolute z-value in logistic regression, and variable importance in other three methods.

Variable rankings in four methods for predicting SGA are shown in Table 6.2. All the four classifiers suggest that *smk*, *pwtgain*3 and *parity* are the most relevant predictors. *ppwtstat*, *prvbig* and *prvlbw* are also highly relevant predictors. *pihyp* is in higher rank in logistic regression and the boosted model, but in lower rank in random forest and does not appear in tree model This is due to the randomization of tree and random forest models. Others are less influential.

Some diagnostic properties for predicting LGA by the four classifiers are shown in Table 6.3.

Comparing with logistic regression for predicting LGA, the boosted tree model has greater accuracy (0.7107) when assessing performance using AUC. Random forest

Prediction model	Threshold	MCE	Sensitivity	Specificity	AUC
Logistic regression	0.153	0.3436	0.63	0.66	0.7085
Classification tree	0.484	0.3385	0.59	0.67	0.6817
Random forest	0.367	0.3363	0.60	0.68	0.6826
Boosting tree	0.155	0.3365	0.63	0.67	0.7107

Table 6.3: Diagnostic properties of four classifiers for predicting LGA. The threshold, the misclassification error rate (MCE), sensitivity, specificity and area under the ROC curve (AUC) are presented.

and tree models have less accuracy than that of logistic regression (0.7085). The tree model has the least accuracy (0.6817). In addition, the boosted tree model has the same sensitivity as that of logistic regression. Random forest and tree models have less sensitivity than logistic regression. The tree model has the least sensitivity (0.59).

Ranking	Logistic regression	Classification tree	Random forest	Boosting tree
1	prvbig	prvbig	smk	prvbig
2	ppwtstat	pwtgain3	prvbig	pwtgain3
3	pwtgain3	smk	pwtgain3	ppwtstat
4	smk	ppwtstat	ppwtstat	smk
5	dm	dm	ses5	matage
6	gdm	parity	matage	dm
7	parity		rural	parity
8	prvlbw		gdm	gdm
9	rural		dm	ses5
10	pihyp		pihyp	prvlbw
11	matage		parity	rural
12	psych		prvlbw	pihyp
13	hyp		prvcs	prvgdm
14			psych	prvcs
15			prvgdm	hyp
16			hyp	psych
17			chabus	chabus

Table 6.4: Variable rankings in four methods for predicting LGA. Variables are ranked from high relevance to low relevance based on absolute z-value in logistic regression, and variable importance in other three methods.

Variable rankings in four methods for predicting LGA are shown in Table 6.4. All the four classifiers suggest that *prvbig*, *pwtgain3*, *ppwtstat*, and *smk* are the most relevant predictors. *dm* is also a highly relevant predictor. Others are less influential.

In fact, none of the more modern techniques : boosting tree, random forest and decision tree did much better than the more traditional logistic regression for this data set.

As to future work, we can combine boosting with the logistic regression model. Or we can use other machine learning methods such as Neural Network and Support Vector Machines, which may lead to higher prediction accuracy.

Bibliography

[1] L. Breiman, J. Friedman, R. Olshen, C. Stone. 1984. *Classification and Regression Trees.* New York: Wadsworth.

[2] L. Breiman. 1996a. "Bagging predictors." *Machine Learning.* 26(2): 123-140.

[3] L. Breiman. 2001. "Random forests." *Machine Learning.* 45: 5-32.

[4] U. G. Das, G. Sysyn. 2004. "Abnormal fetal growth: intrauterine growth retardation, small for gestational age, large for gestational age." *Pediar Clin North Am.* 51: 639-54, viii.

[5] S. R. DeVader, H. L. Neeley, T. D. Myles, T. L. Leet. 2007. "Evaluation of gestational weight gain guidelines for women with normal prepregnancy body mass index." *Obstet Gynecol.* 110: 745-751.

[6] T. Dietterich. 1998. "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting and randomization". *Machine Learning.* 1-22.

[7] Y. Freund. 1995. "Boosting a weak learning algorithm by majority." *Information and Computation* 121(2): 256-285.

[8] Y. Freund, R. Schapire. 1996a. "Experiments with a new boosting algorithm." *Machine Learning: Proceedings of the Thirteenth International Conference.* Morgan Kauffman, San Francisco. 148-156.

[9] J. Friedman. 2001. "Greedy function approximation: A gradient boosting machine." *Annals of Statistics* 29(5): 1189-1232.

[10] J. Friedman, T. Hastie, R. Tibshirani. 2000. "Additive logistic regression: a statistical view of boosting (with discussion)." *Annals of Statistics* 28: 337-307.

[11] T. Hastie, R. Tibshirani, J. Friedman. 2009. *The Elements of Statistical Learning*, 2nd Edition. New York: Springer.

[12] P. Huber. 1964. "Robust estimation of a location parameter." *Annals of Mathematical Statistics* 53: 73-101.

[13] D. Hosmer, S. Lemeshow. 2000. *Applied Logistic Regression*, 2nd Edition. New York: Wiley.

[14] G. James, T. Hastie, D. Witten, R. Tibshirani. 2013. *An Introduction to Statistical Learning, with Applications in R.* New York: Springer.

[15] M. S. Kramer, R. W. Platt, S. W. Wen, K. S. Joseph, A. Allen, M. Abrahamowicz, B. Blondel, G. Breart. 2001. "A new and improved population-based Canadian reference for birth weight for gestational age." *Pediatrics* 108: E35.

[16] S. Longo, L. Bollani, L. Decembrino, A. D. Comite, M. Angelini, M. Stronati. "Short-term and long-term sequelae in intrauterine growth retardation (IUGR)." *J Matern Fetal Neonatal Med.*

[17] V. K. Minior, M. Y. Divon. 1998. "Fetal growth restriction at term: myth or reality?" *Obstet Gynecol.* 92: 57-60.

[18] E. A. Nohr, M. Vaeth, J. L. Baker, T. I. Sorensen, J. Olsen, K. M. Rasmussen. 2008. "Combined associations of prepregnancy body mass index and gestational weight gain with the outcome of pregnancy." *Am J Clin Nutr.* 87: 1750-1759.

[19] K. M. Rasmussen, A. L. Yaktine. *Weight gain during pregnancy: reexamining the guidelines.* 2009. Washington (DC): National Academies Press.

[20] R. Schapire. 1990. "The strength of weak learnability." *Machine Learning* 5(2): 197-227.

[21] R. Schapire, Y. Singer. 1999. "Improved boosting algorithms using confidence-rated predictions." *Machine Learning* 37(3): 297-336.

[22] A. M. Siega-Riz, M. Viswanathan, M. K. Moos, A. Deierlein, S. Mumford, J. Knaack, P. Thieda, L. J. Lux, K. N. Lohr. 2009. "A systematic review of outcomes of maternal weight gain according to the Institute of Medicine recommendations: birthweight, fetal growth, and postpartum weight retention." *Am J Obstet Gynecol.* 201: 339.e1-339.14.

[23] F. A. Van Assche, K. Holemans, L. Aerts. 2001. "Long-term consequences for offspring of diabetes during pregnancy." *Br Med Bull.* 60: 173-182.