Prediction of Apgar Score Using Statistical Learning

by

Nina Oryshkewych

BS, The Ohio State University, 2019

Submitted to the Graduate Faculty of the

Department of Biostatistics

Graduate School of Public Health in partial fulfillment

of the requirements for the degree of

Master of Science

University of Pittsburgh

2022

UNIVERSITY OF PITTSBURGH

GRADUATE SCHOOL OF PUBLIC HEALTH

This thesis was presented

by

Nina Oryshkewych

It was defended on

April 25, 2022

and approved by

Jeanine M. Buchanich, MEd, MPH, PhD, Vice Chair for Practice & Research Associate Professor, Department of Biostatistics, School of Public Health, University of Pittsburgh

Jenna C. Carlson, PhD, Assistant Professor, Departments of Biostatistics and Human Genetics, School of Public Health, University of Pittsburgh

Evelyn O. Talbott, DrPH, MPH, Professor, Department of Epidemiology, School of Public Health, University of Pittsburgh

Ada O. Youk, PhD, Associate Professor & Vice Chair of Education, Department of Biostatistics, School of Public Health, University of Pittsburgh

Thesis Advisor: Jeanine M. Buchanich, MEd, MPH, PhD, Vice Chair for Practice & Research Associate Professor, Department of Biostatistics, School of Public Health, University of Pittsburgh

Prediction of Apgar Score Using Statistical Learning

Nina Oryshkewych, MS

University of Pittsburgh, 2022

Background: Apgar score is a measure of neonatal health. A low Apgar score has been linked to several adverse health outcomes. Ambient air pollution has been shown to be a major threat to public health, but there is limited research on the relationship between maternal exposure to air pollution and Apgar score.

Methods: Maternal exposure to air pollution was calculated for each trimester and for each of the seven criteria air pollutants based on the nearest monitor to each mother's residence. A combination of random over- and under-sampling was performed on the training data to balance the class distribution of Apgar score. Extreme gradient boosting (XGBoost) and logistic regression were used to build eight classification models – two using all predictors and six trimester-specific models.

Results: All models had poor discriminative ability. The best performing model was the XGBoost second trimester model, with an AUC of 0.627. In the XGBoost models, gestational age appeared to be the most important predictor of Apgar score, followed by the air pollution exposure variables. In the logistic regression models, gestational age was the most significant predictor.

Conclusion: Gestational age is the primary driver of Apgar score, and exposure to air pollution may be important as well. While none of the models had adequate predictive ability, there are a few limitations to this study that may have hindered their performance. Future

research should consider more sophisticated resampling techniques as well as geospatial modelling of pollution concentrations in order to improve the quality of the data.

Public Health Significance: While many studies have investigated the consequences of a low Apgar score, existing research lacks in exploration of factors that influence Apgar score. This study suggests the possibility that exposure to ambient air pollution could be linked to a low five minute Apgar score. A classification model for Apgar score could guide practitioners and public health officials in implementing preventative measures to protect neonatal health.

Table of Contents

List of Tables

List of Figures

1.0 Introduction

1.1 Apgar Score

Apgar score is a measure is a measure of newborn health. It is scored 0-10 and is comprised of five components, each of which are scored 0-2 and summed to generate a total score. The components include: breathing effort, heart rate, muscle tone, grimace response or reflex irritability, and color. Scores of 7-10 are generally considered normal, while scores of 4-6 are considered moderately abnormal, and scores of 0-3 are considered low. Apgar scores are measured one minute after birth and again five minutes after birth. Any infants who score less than 7 or require resuscitation at five minutes are further measured at 5-minute intervals (Simon, Hashmi, & Bragg, 2021).

The Apgar score was originally developed as a metric to determine whether an infant required resuscitation. Current guidelines, however, state that resuscitation must be initiated for infants who require it before the 1-minute Apgar score is measured. Nonetheless, the American College of Obstetricians and Gynecologists and the American Academy of Pediatrics maintain Apgar scoring as an accepted method of assessing infant health (Simon et al., 2021). It remains a useful tool in detecting signs of cardiovascular or respiratory complications.

1.1.1 Implications

An Apgar score alone cannot be used to predict a newborn's health trajectory; however, a number of studies suggest that infants with low Apgar scores are at higher risk for certain complications. At one minute, a low Apgar score is not necessarily indicative of any adverse outcomes ("Committee Opinion No. 644: The Apgar Score," 2015). Though at five minutes, there is substantial evidence of an association with low Apgar scores and adverse health outcomes. An Apgar score of 0-3 at five minutes has been shown to be associated with increased risk for neonatal mortality (Li et al., 2013). Furthermore, there is evidence of an association between a low 5-minute Apgar score and development of cerebral palsy. A study of over 200,000 newborns found that the risk for neonatal death in infants who scored 0-3 increased 386-fold compared to infants who scored 7-10; the risk for developing cerebral palsy increased by 81-fold (Moster, Lie, Irgens, Bjerkedal, & Markestad, 2001). Additionally, infants with abnormal Apgar scores are at an increased risk of developing neurologic disabilities even many years after birth. While the relative risks of disability are considerable for newborns with low Apgar scores, it is important to note, however, that most of low-scoring infants who survive do not end up developing disabilities (Ehrenstein, 2009).

1.2 Air Pollution

1.2.1 Criteria Air Pollutants

The EPA classifies 6 common pollutants as “criteria air pollutants”: carbon monoxide, lead, nitrogen dioxide, ground-level ozone, particulate matter, and sulfur dioxide. These pollutants are subject to National Ambient Air Quality Standards, which were set by the Clean Air Act. This ordinance defines two types of standards – primary and secondary. Primary standards are meant to protect public health, especially for populations that are sensitive to air pollution (i.e. asthma patients, children, and the elderly). Secondary standards provide a broader range of protection; they are meant to prevent poor visibility and harm to animals, agriculture, and buildings. The Clean Air Act has made a considerable impact on air pollution levels in the United States. Since 1990, emissions of major air pollutants have consistently decreased; since 2000, the number of unhealthy air quality days across 35 major US cities has decreased by 62% (US Environmental Protection Agency, 2019). Despite the major improvements that have been made in air quality, in 2019 nearly 82 million people across the Unites States lived in counties that exceeded NAAQS primary standards (US Environmental Protection Agency, 2020).

1.2.2 Health Effects

According to the World Health Organization, 4.2 million deaths across the globe each year can be attributed to ambient air pollution; 99% of the global population experiences air quality conditions that exceed WHO guidelines (World Health Organization, 2021). Furthermore, the research team at the Global Burden of Disease project estimate that air pollution accounts for a fifth of neonatal mortality worldwide and that that nearly 500,000 neonatal deaths in 2019 could be attributed to air pollution (Health Effects Institute, 2020). While significant progress continues to be made, it is evident that air pollution remains a major threat to public health.

1.2.3 Air Quality in Allegheny County

Southwest Pennsylvania has a long history of polluted air. While considerable progress has been made in recent years, Allegheny County still suffers from poor air quality. According to the American Lung Association, Allegheny county is the 16th most polluted county in the United States based on annual levels of particulate matter (American Lung Association, 2021). Furthermore, as seen in Figure 1, EPA data from 2009-2020 shows that the majority of days in this time period in Allegheny county had a daily air quality index (AQI) that exceeded the "good" threshold (US Environmental Protection Agency, 2021).

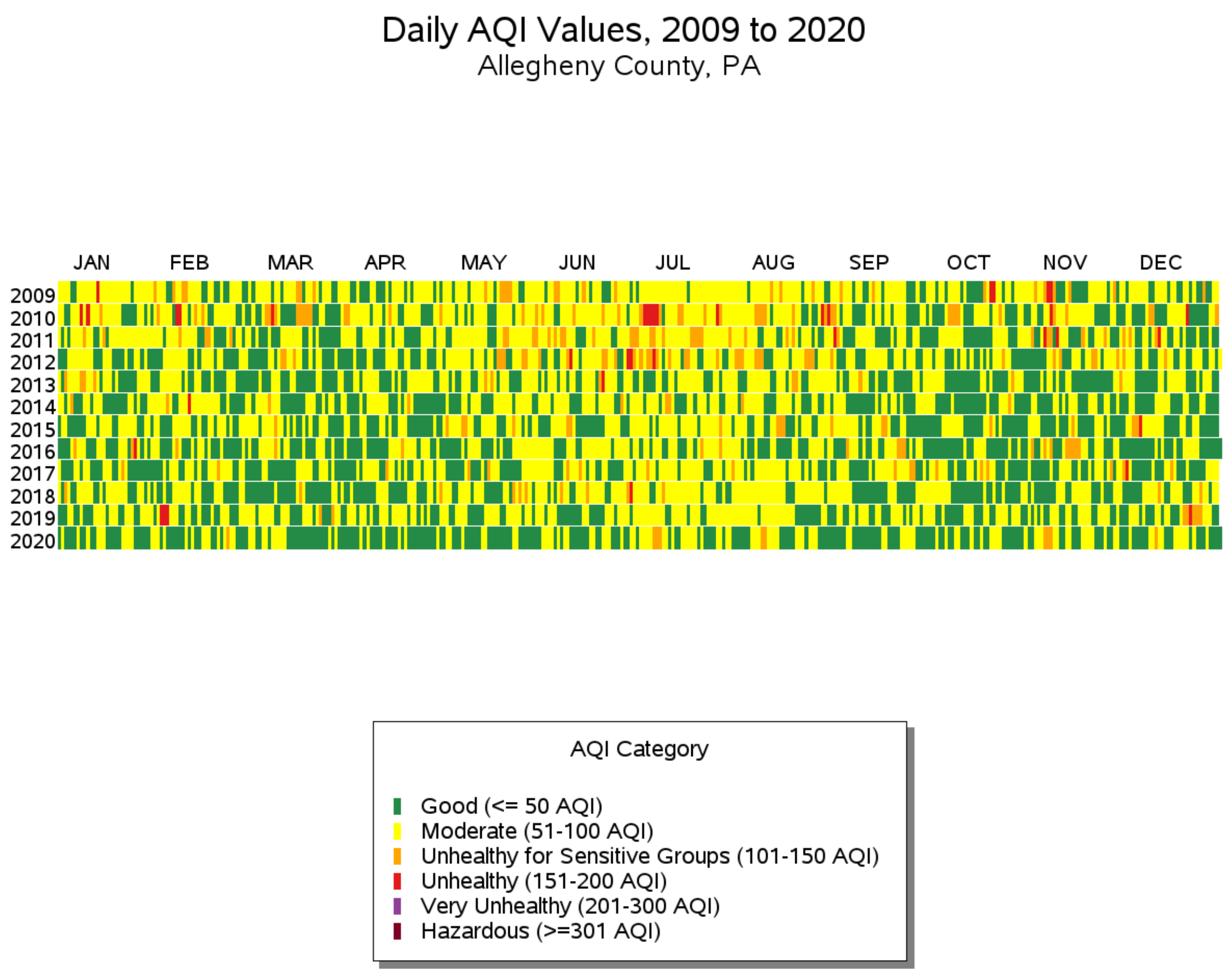

Figure 1: Allegheny County AQI Values

1.3 Objectives

While a number of studies have linked air pollution to detrimental health effects, limited research has been done on the relationship between air pollution and Apgar score. The aims of this thesis are to construct and compare several models that predict Apgar score and to determine whether maternal exposure to criteria air pollutants is important in classifying a score as normal or abnormal. Given the implications that an Apgar score has on an infant's, the magnitude of the effect that air pollution has on public health, such a model would contribute meaningfully to existing research on neonatal health. Furthermore, Allegheny county's air quality continues to be a significant public health issue; this study has the potential to uncover further evidence of the effects of Allegheny county's air quality on health.

2.0 Methods

2.1 Variables of Interest

Five-minute Apgar score was designated as the outcome variable. The exposure variables of interest included average ambient concentrations of each of the criteria pollutants by trimester. Additional predictors included: child's sex, season of birth, gestational age, mother's age, mother's BMI, diagnosis of gestational diabetes, maternal race, maternal ethnicity, paternal race, paternal ethnicity, maternal education, number of cigarettes smoked prior to pregnancy, and number of cigarettes smoked during pregnancy.

2.2 Data Cleaning and Management

All birth-related data were acquired from the Pennsylvania Department of Health. The data spanned the years 2010-2020 and included births from Allegheny County. Records that had multiple births, infants weighing less than 500 g, mothers older than 45, a gestational age less than 22 weeks, or any unknown or missing variables were dropped from the analysis. Due to lack of variability, the categories of certain variables were collapsed. Maternal and paternal race were collapsed into White, Black/African American, and Other. Mother's education was collapsed into less than high school, high school or GED, some college, Bachelor's degree, and graduate degree. In addition, the number of cigarettes smoked during each trimester was summed into a single variable – total number of cigarettes smoked during pregnancy. Lastly, Apgar score was categorized into two groups – abnormal or normal; scores of 0-6 were included in the abnormal category and scores of 7-10 were included in the normal category, as suggested by Simon et al. (Simon et al., 2021).

Air quality data for Allegheny County and surrounding counties were downloaded from the EPA Air Data page (US Environmental Protection Agency, 2021); these data spanned the years 2009-2020. Daily concentration summaries were acquired for the following pollutants: carbon monoxide, nitrogen dioxide, ozone, lead, PM_{10}, $PM_{2.5}$, and sulfur dioxide. Assigning air pollution exposure required several steps. First, we estimated the start of gestation using the gestational age, as well as cutoffs dates for each trimester. The beginning of the second trimester was estimated to start thirteen weeks after the start of gestation and the beginning of the third trimester was estimated to start 26 weeks after the start of gestation. Next, we identified the monitoring sites for each pollutant that were active during each gestational period, noting whether certain monitors became active or inactive throughout different trimesters. Next, for each trimester, we found the monitor nearest to the mother's residence. We allowed for monitors to differ by trimester depending on activity status throughout the pregnancy. In addition, we considered the possibility that some mothers may have lived closer to monitoring sites in neighboring counties, and thus included monitors from each of Allegheny's border counties in the assignment process. Lastly, the average pollution concentration from the closest monitor was averaged for the duration of each trimester. Figure 1 illustrates the step-by-step process of exposure assignment.

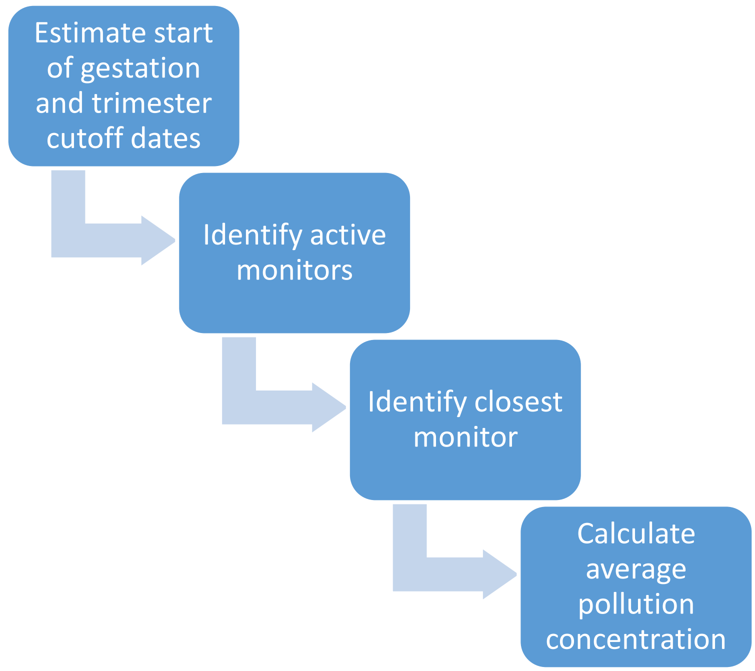

Figure 2: Exposure Assignment for Each Birth and Each Pollutant by Trimester

2.3 Model Pre-Processing

After data preparation was complete, all categorical variables were converted to numeric variables using dummy encoding. 70% of the data was randomly selected to be used as a training set for model building and the remaining 30% was held out to test the performance of the models. Because the class distribution of Apgar scores was highly imbalanced, the training data was resampled to balance the outcome classes. A combination of random under-sampling of the majority class and random over-sampling of the minority class was performed until the classes became approximately equal, while maintaining the original sample size. The 'ROSE' package in R was used to implement this. This was a necessary step in data preparation, as most machine learning techniques rely on a balanced outcome to build reliable models. If classes are highly imbalanced, a model will lose discrimination capability – it will tend to incorrectly predict instances as belonging to the majority class in order to maximize overall accuracy.

2.4 Model Building

Extreme gradient boosting and logistic regression were used to build several classification models. The first models were built using air pollution exposure variables for all three trimesters. Next, three sets of trimester-specific models were built.

2.4.1 Extreme Gradient Boosting

Gradient boosting is an ensemble learning technique, in which a large quantity of decision trees are formed through an iterative process, where each iteration tweaks the previous model in an attempt to correct prior misclassifications. The optimal model is constructed by minimizing a loss function; in the case of classification, this function is the negative binomial log-likelihood (Friedman, 2001):

$$L(y, F) = log\left(1 + exp(-2yF)\right),\ y \in \{-1, 1\}, \text{(Eq. 1)}$$

$$\text{where } F(\boldsymbol{x}) = \frac{1}{2}\log\left[\frac{\Pr(y=1\ |\boldsymbol{x})}{\Pr(y=-1\ |\boldsymbol{x})}\right] \text{(Eq. 2)}$$

Gradient boosting is implemented in R's 'caret' package, which integrates functions from the 'xgboost' package, which carries out a version of gradient boosting known as extreme gradient boosting (XGBoost). The XGBoost algorithm carries out the principles of gradient boosting, while applying additional regularization to prevent over-fitting. The function that builds the model intakes several hyperparameters, which are tuned to maximize model performance. The hyperparameters available for tuning are: number of iterations, maximum tree depth, learning rate, gamma, column sample, minimum node size, and subsample. The number of iterations refers to the number of decision trees that are constructed. The learning rate is a shrinkage parameter – it scales down the contribution of each tree that is added to the model. Gamma, minimum child weights, and maximum depth are all used to control tree complexity. Gamma refers to the minimum reduction of the loss function that is required to create an additional partition in the tree. The minimum child weight is the minimum number of births that are allowed in a leaf node of the decision tree. The maximum depth restricts the number of partitions that can be made from root to lead. The subsample indicates what proportion of the training data should be sampled to grow each tree. The column sample indicates what proportion of features should be sampled to build each tree. A sequence of possible values are inputted for each hyperparameter; each possible combination of hyperparameters is entered as a row in a tuning grid, which is then searched for the combination that produces the best performing model. The default tuning grid in the 'caret' package was searched via five-fold cross-validation for the optimal combination of parameters.

2.4.1.1 Feature Importance

Variable importance was assessed for each gradient boosted classifier using two different metrics – gain and cover. As implemented by 'xgboost', variable importance measured by gain describes the average information gain attributed to each feature out of all the trees that were constructed, whereas cover describes the relative number of observations related to each feature (Chen et al., 2022)

2.4.1.2 Partial Dependence

Partial dependence plots were constructed for the top five important variables ranked by gain for the full XGBoost model. The purpose of these plots is to visualize the marginal effect of individual variables on the predicted probability of an abnormal Apgar score.

2.4.2 Logistic Regression

Logistic regression is a model used for binary outcomes, where each predictor receives a coefficient that contributes to the prediction of the response variable. Logistic regression is a type of generalized linear model, in which the outcome variable follows the distribution within the exponential family; in the case of logistic regression, the logit function links the expected value of the outcome to the covariates and their coefficients. The theory behind this model is as follows.

Let $P(Y_i = 1|X_i) = p_i$ and $P(Y_i = 0|X_i) = 1 - p_i$

The model equation then becomes:

$$logit(p_i) = log\left(\frac{p_i}{1-p_i}\right) = X_i\beta \text{ (Eq. 3)}$$

and

$$p_i = E(Y_i|X_i) = \frac{exp(X_i\beta)}{1+exp(X_i\beta)} \text{ (Eq. 4)}$$

Logistic regression is also implemented in the ‘caret’ package, which utilizes functions from the ‘glm’ package. This model does not incorporate any hyperparameters. The final model output was optimized via five-fold cross-validation.

2.5 Model Evaluation

Model performance was evaluated using several metrics: accuracy, sensitivity, specificity, positive predictive value, negative predictive value, the area under the receiver operating characteristic (ROC) curve, no information rate. Overall model accuracy was measured by calculating the proportion of Apgar scores that were correctly predicted out of the entire testing set. To calculate sensitivity and specificity, a confusion matrix was constructed, in which an abnormal Apgar score was considered the "positive class". A confusion matrix, shown in Table 1, classified predictions into one of 4 categories: true positive, true negative, false positive, and false negative.

Table 1: Confusion Matrix

		Reference	
		Abnormal	*Normal*
Prediction	*Abnormal*	True Positive (TP)	False Positive (FP)
	Normal	False Negative (FP)	True Negative (TN)

Sensitivity is the true positive rate – it refers to the probability of a case being predicted as positive, given the case is truly positive; in the context of our problem, this means the probability of the model predicted an abnormal Apgar score, given that the score is truly abnormal. Specificity is the true negative rate and describes the probability of a case being predicted as negative, given the case is truly negative; in terms of the research question, this refers to the probability of the classifier predicting a normal Apgar score, given that the score is truly normal. The no information rate refers to the proportion of that total sample that belongs to the majority class; in other words, it describes the probability of correctly classifying an observation by simply predicting it to be of the majority class. In order to determine the overall significance of the models, a one-sided hypothesis test was conducted to assess whether model accuracy was greater than the no information rate. Table 2 summarizes all model evaluation metrics that were used.

Table 2: Performance Metrics

Metric	Formula
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$
Sensitivity	$\frac{TP}{TP + FN}$
Specificity	$\frac{TN}{TN + FP}$
Positive Predictive Value	$\frac{TP}{TP + FP}$
Negative Predictive Value	$\frac{TN}{TN + FN}$
No Information Rate	$\frac{n_{majority\ class}}{n_{total}}$

The receiver operating characteristic (ROC) curve takes both sensitivity and specificity into account, creating a more balanced measure of model performance. The ROC curve is computed by plotting the sensitivity against 1- the specificity; the area under the curve (AUC) quantifies the strength of the discriminative ability of the model. The AUC was used to identify the best model.

3.0 Results

3.1 Summary Statistics

The original birth record dataset contained 141,613 observations. After applying exclusion criteria and dropping records with missing values, the resulting data set contained 61,118 observations. Table 3 shows summary statistics of all variables directly related to the newborns and their parents.

Table 3: Summary Statistics

	Births
Sex	
F	30043 (49.2%)
M	31075 (50.8%)
Season of Birth	
Fall	15922 (26.1%)
Spring	15771 (25.8%)
Summer	15380 (25.2%)
Winter	14045 (23.0%)
Gestational Age (weeks)	
Mean (SD)	38.9 (1.59)
Median [Min, Max]	39.0 [26.0, 45.0]
Mother's Age	
Mean (SD)	30.2 (5.14)
Median [Min, Max]	30.0 [13.0, 45.0]
BMI	

	Births
Normal	32182 (52.7%)
Obese	12533 (20.5%)
Overweight	14571 (23.8%)
Underweight	1832 (3.0%)
Gestational Diabetes	
No	57926 (94.8%)
Yes	3192 (5.2%)
Maternal Race	
All other races	3774 (6.2%)
Black or African American	7587 (12.4%)
White	49757 (81.4%)
Maternal Ethnicity	
Hispanic	1119 (1.8%)
Not Hispanic	59999 (98.2%)
Paternal Race	
White	47883 (78.3%)
Black	9311 (15.2%)
All other races	3924 (6.4%)
Paternal Ethnicity	
Not Hispanic	59882 (98.0%)
Hispanic	1236 (2.0%)
Maternal Education	
Bachelor's degree	19721 (32.3%)
Graduate degree	15006 (24.6%)
High school or GED	8612 (14.1%)
Less than high school	2200 (3.6%)
Some college	15579 (25.5%)
Number of Cigarettes Smoked Prior to Pregnancy	
Mean (SD)	1.74 (5.68)

	Births
Median [Min, Max]	0 [0, 98.0]
Number of Cigarettes Smoked During Pregnancy	
Mean (SD)	2.33 (9.36)
Median [Min, Max]	0 [0, 294]
5-Minute Apgar Score	
Abnormal	631 (1.0%)
Normal	60487 (99.0%)

Table 4 summarizes maternal exposure to each of the criteria air pollutants by trimester. Average exposure appeared to remain fairly consistent across trimesters.

Table 4: Air Pollution Exposure

	Mean (SD)		
	First Trimester	Second Trimester	Third Trimester
Carbon Monoxide (ppm)	0.310 (0.113)	0.311 (0.114)	0.313 (0.115)
Nitrogen Dioxide (ppb)	10.3 (3.61)	10.3 (3.65)	10.2 (3.72)
Ozone (ppm)	0.0291 (0.00711)	0.0292 (0.00725)	0.0293 (0.00705)
Lead ($\mu g/m^3$)	0.0126 (0.0274)	0.0118 (0.0266)	0.0116 (0.0267)
$PM_{2.5}$ ($\mu g/m^3$)	10.3 (2.42)	10.3 (2.47)	10.1 (2.33)
PM_{10} ($\mu g/m^3$)	17.2 (4.77)	17.2 (4.80)	17.2 (4.83)
Sulfur Dioxide (ppb)	2.17 (1.99)	2.12 (1.98)	2.04 (1.95)

The class distribution of Apgar score was heavily imbalanced – with 42,352 newborns having a normal Apgar score and only 431 newborns having an abnormal Apgar score in the training data set. Applying a combination of random oversampling of the minority class and random under-sampling of the majority class produced a much more balanced outcome distribution – with 21,342 normal Apgar scores and 21,441 abnormal Apgar scores (Table 5).

Table 5: Outcome Class Distribution

	Original Data	Re-sampled Data
Abnormal	431	21,441
Normal	42,352	21,342

3.2 Air Quality Monitors

The number of monitors for each pollutant in each county can be seen in Table 6. The study area contained considerably more monitors for particulate matter than for the other criteria pollutants. Maps of the monitors locations can be found in Appendix A.

Table 6: Air Quality Monitors

	Carbon Monoxide	Lead	Nitrogen Dioxide	Ozone	PM_{10}	$PM_{2.5}$	Sulfur Dioxide
Allegheny	5	4	5	5	11	10	7
Armstrong	0	0	0	1	0	1	0
Beaver	0	5	1	3	1	1	2
Butler	0	1	0	0	0	0	0
Washington	2	0	2	4	1	4	2
Westmoreland	0	2	0	2	0	1	0

3.3 Model Output

3.3.1 XGBoost

3.3.1.1 Tuning

A total of 108 combinations of hyperparameters were tested for each XGBoost model. The results of the tuning process for the full model can be seen in Figure 2; the overall predictive accuracy in the training set was compared for each combination of hyperparameters that was tested. A tree depth of 3 consistently yielded substantially higher accuracy than shorter tree depths. Furthermore, accuracy also increased as the number of boosting iterations increased.

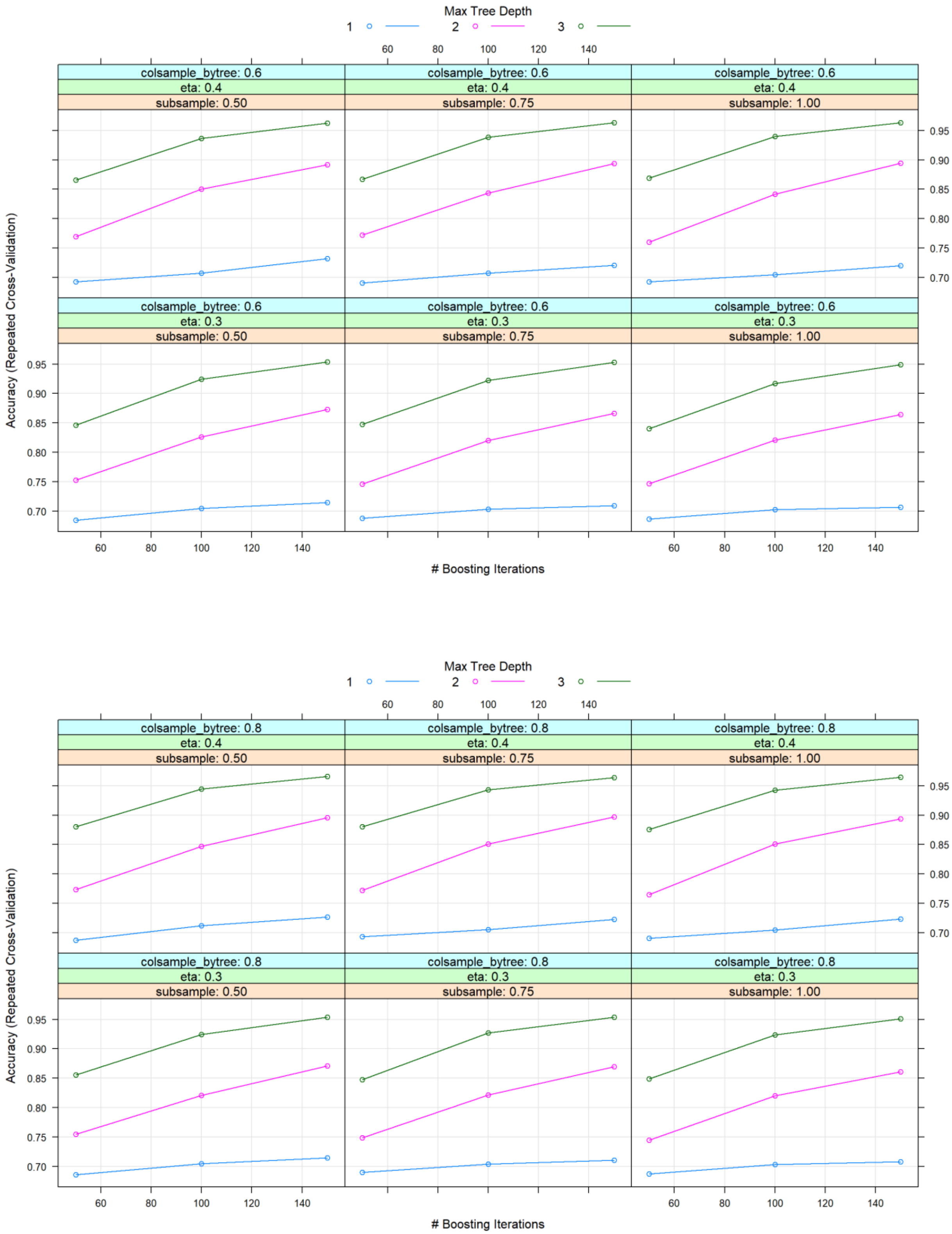

Figure 3: Tuning Results

The best tune of the full model is shown in Table 7. This combination of hyperparameters resulted in an accuracy of 0.966 in the training set.

Table 7: Best Tune

Learning Rate	Maximum Tree Depth	Gamma	Column Sample	Minimum Node Size	Subsample	Number of Iterations	Accuracy
0.4	3	0	0.8	1	0.5	150	0.966

3.3.1.2 Feature Importance

Feature importance was extracted from each of the XGBoost models. Figures 4 displays the top 20 features ranked by gain for each XGBoost model that was built. In each of the models, gestational age was by far the most important predictor of Apgar score. Furthermore, the air pollution exposure variables consistently ranked higher in importance than demographic variables.

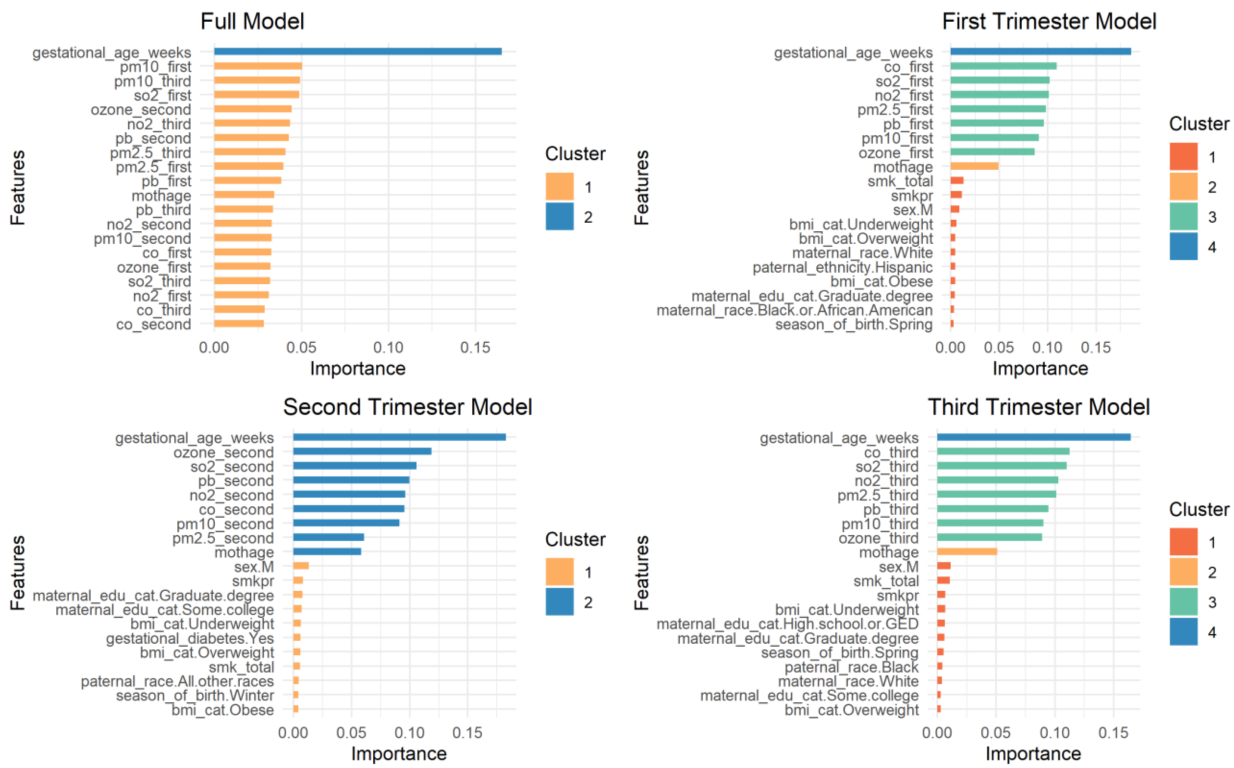

Figure 4: Feature Importance - Gain

Figures 5 displays the top 20 features of each XGBoost model ranked by cover. From this perspective, gestational age was only the most important feature in the full model. However, the air pollution exposure variables still tended to outrank the demographic variables.

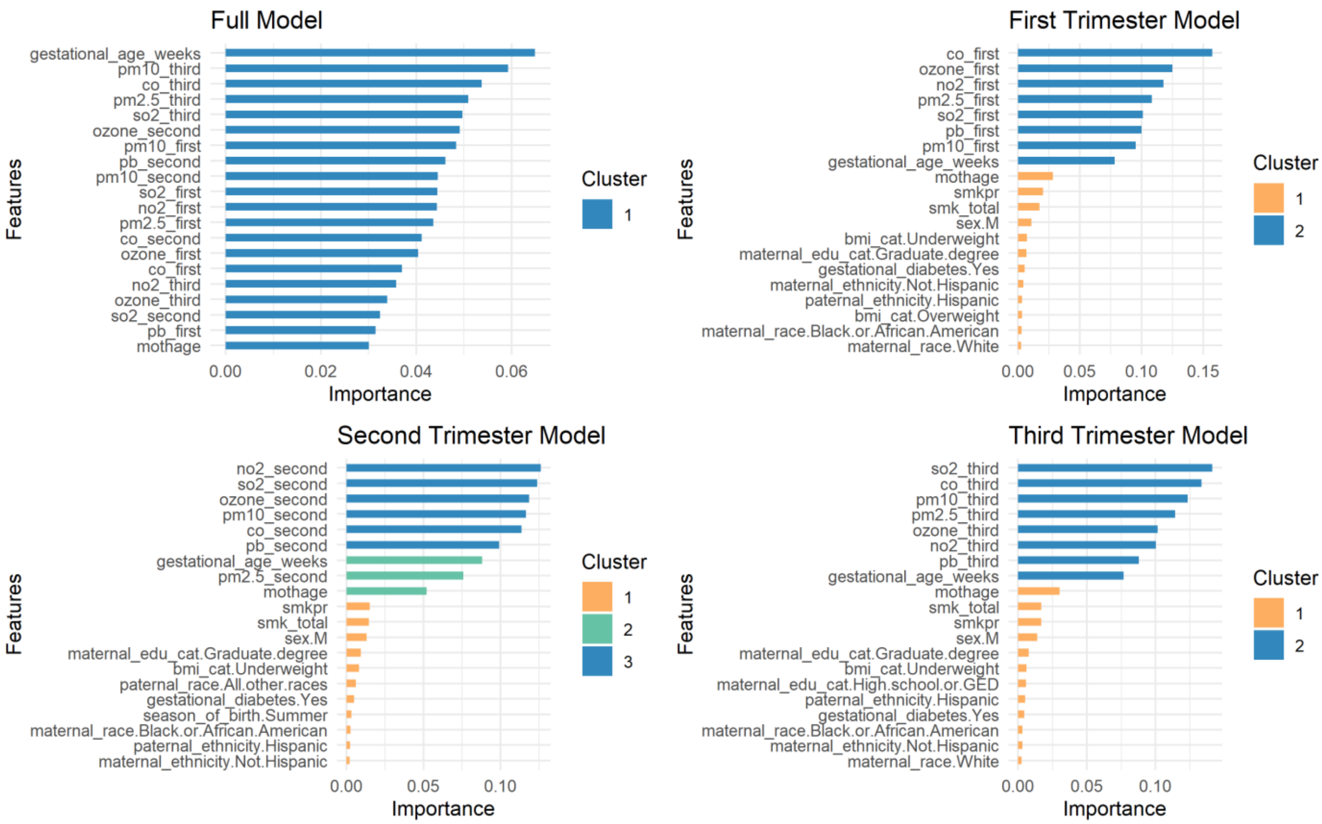

Figure 5: Feature Importance - Cover

3.3.1.3 Partial Dependence

Figure 6 displays the partial dependence of Apgar score on the top 5 most important predictors measured by gain. The partial dependences are congruent with what was seen with variable importance – the probability of an abnormal Apgar score appears to vary the most depending on gestational age, whereas the probability changes much less depending on pollutant concentrations.

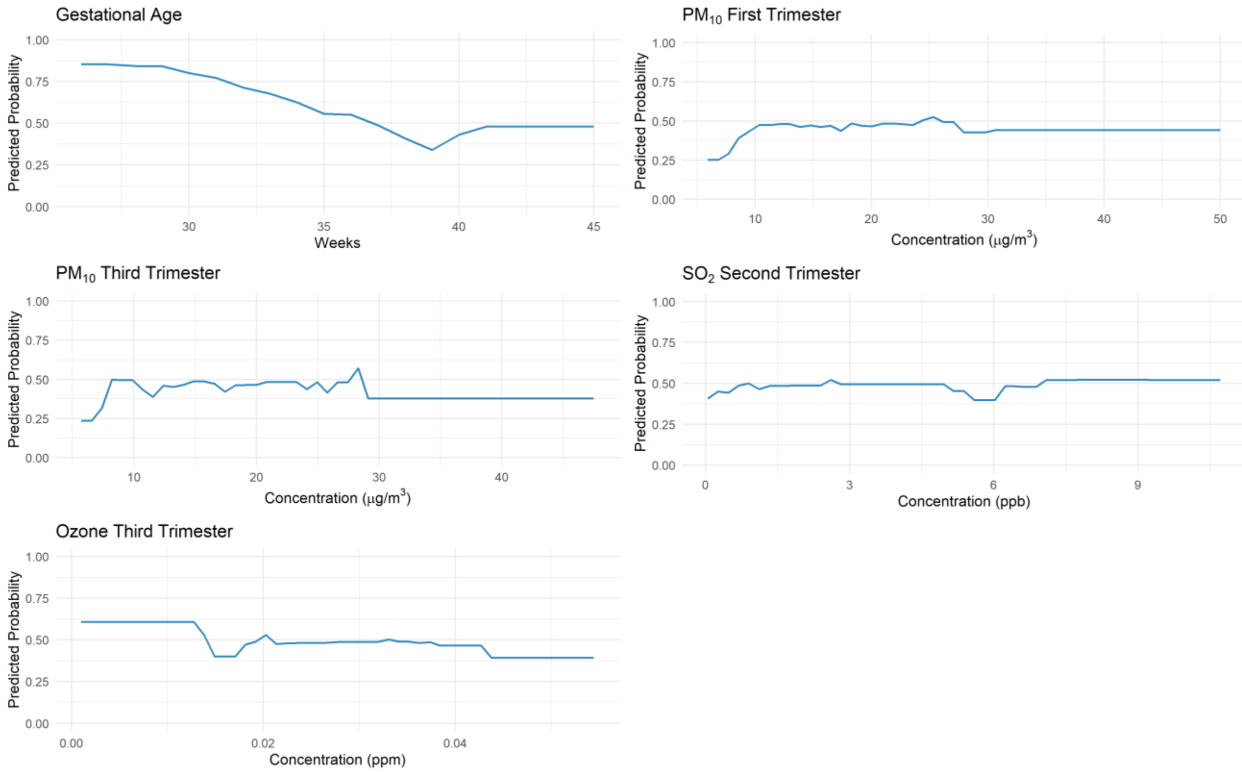

Figure 6: Partial Dependence Plots

3.3.2 Logistic Regression

The output of the logistic regression is shown in Table 8. A majority of the predictors had statistically significant coefficients. Similar to the XGBoost models, gestational age was the most significant predictor of Apgar score. Output for the trimester-specific logistic regression models can be found in Appendix B.

Table 8: Full Logistic Regression Model Output

	Estimate	Standard Error	z value	Pr(>|z|) Signif.
(Intercept)	9.483	0.267	35.457	0.0000 ***
sex.M	0.333	0.021	15.867	0.0000 ***
season_of_birth.Spring	-0.249	0.054	-4.640	0.0000 ***
season_of_birth.Summer	-0.133	0.046	-2.869	0.0041 **
season_of_birth.Winter	-0.060	0.043	-1.387	0.1653
gestational_age_weeks	-0.219	0.005	-48.399	0.0000 ***
mothage	0.002	0.002	1.085	0.2777
bmi_cat.Obese	-0.011	0.028	-0.382	0.7022
bmi_cat.Overweight	0.017	0.026	0.672	0.5013
bmi_cat.Underweight	-0.819	0.074	-11.132	0.0000 ***
gestational_diabetes.Yes	0.103	0.044	2.340	0.0193 *
maternal_race.Black.or.African.American	0.532	0.076	6.954	0.0000 ***
maternal_race.White	0.227	0.064	3.524	0.0004 ***
maternal_ethnicity.Not.Hispanic	0.159	0.086	1.848	0.0646 .
maternal_edu_cat.Graduate.degree	-0.248	0.028	-8.719	0.0000 ***
maternal_edu_cat.High.school.or.GED	-0.021	0.037	-0.567	0.5704
maternal_edu_cat.Less.than.high.school	-0.147	0.063	-2.318	0.0204 *
maternal_edu_cat.Some.college	-0.041	0.029	-1.381	0.1674
smkpr	0.016	0.002	7.446	0.0000 ***
smk_total	-0.002	0.001	-1.581	0.1140
paternal_race.Black	-0.037	0.049	-0.751	0.4524
paternal_race.All.other.races	0.093	0.061	1.514	0.1300
paternal_ethnicity.Hispanic	-0.462	0.083	-5.572	0.0000 ***
co_first	-0.424	0.151	-2.808	0.0050 **
co_second	-0.757	0.183	-4.133	0.0000 ***
co_third	0.126	0.143	0.879	0.3794
no2_first	-0.021	0.006	-3.458	0.0005 ***
no2_second	-0.029	0.007	-3.982	0.0001 ***
no2_third	-0.015	0.006	-2.351	0.0187 *
ozone_first	-22.279	2.821	-7.896	0.0000 ***
ozone_second	-15.492	3.077	-5.035	0.0000 ***
ozone_third	11.602	2.927	3.964	0.0001 ***
pb_first	-3.740	0.549	-6.808	0.0000 ***

	Estimate	Standard Error	z value	$Pr(>\lvert z\rvert)$	Signif.
pb_second	1.288	0.560	2.298	0.0215	*
pb_third	1.211	0.510	2.373	0.0176	*
pm10_first	0.034	0.004	7.922	0.0000	***
pm10_second	-0.049	0.005	-9.824	0.0000	***
pm10_third	0.004	0.005	0.981	0.3268	
pm2.5_first	0.024	0.008	2.982	0.0029	**
pm2.5_second	0.061	0.008	8.119	0.0000	***
pm2.5_third	-0.025	0.008	-2.972	0.0030	**
so2_first	-0.098	0.012	-8.402	0.0000	***
so2_second	0.037	0.013	2.826	0.0047	**
so2_third	-0.098	0.012	-8.088	0.0000	***

Signif. codes: 0 <= '***' < 0.001 < '**' < 0.01 < '*' < 0.05 < '.' < 0.1 < '' < 1

(Dispersion parameter for binomial family taken to be 1)
Null deviance: 5.931e+04 on 42782 degrees of freedom
Residual deviance: 5.366e+04 on 42739 degrees of freedom

3.4 Model Performance

3.4.1 Full Models

All possible predictors were used to fit an extreme gradient boosted (XGBoost) classification model and a logistic regression. As seen in Table 9, the XGBoost classifier correctly predicted 36 abnormal Apgar Scores and 16,748 normal Apgar scores. The GLM classifier correctly predicted 100 abnormal Apgar scores and 12,200 normal Apgar scores, as shown in Table 10.

Table 9: Confusion Matrix: XGBoost Full Model

		Reference	
		Abnormal	*Normal*
Prediction	*Abnormal*	36	1,387
	Normal	164	16.748

Table 10: Confusion Matrix: GLM Full Model

		Reference	
		Abnormal	*Normal*
Prediction	*Abnormal*	100	5,935
	Normal	100	12,200

Additional performance metrics can be seen in Table 7. Neither model had an accuracy that was significantly greater than the no information rate (Table 7).

Table 7: Performance Statistics

	Sensitivity	Specificity	PPV	NPV	Accuracy	NIR	P(Acc > NIR)
XGBoost	0.180	0.923	0.025	0.990	0.915	0.989	1
GLM	0.500	0.673	0.017	0.992	0.671	0.989	1

The logistic regression slightly outperformed the XGBoost classifier, with AUCs of 0.618 and 0.592, respectively (Figure 7).

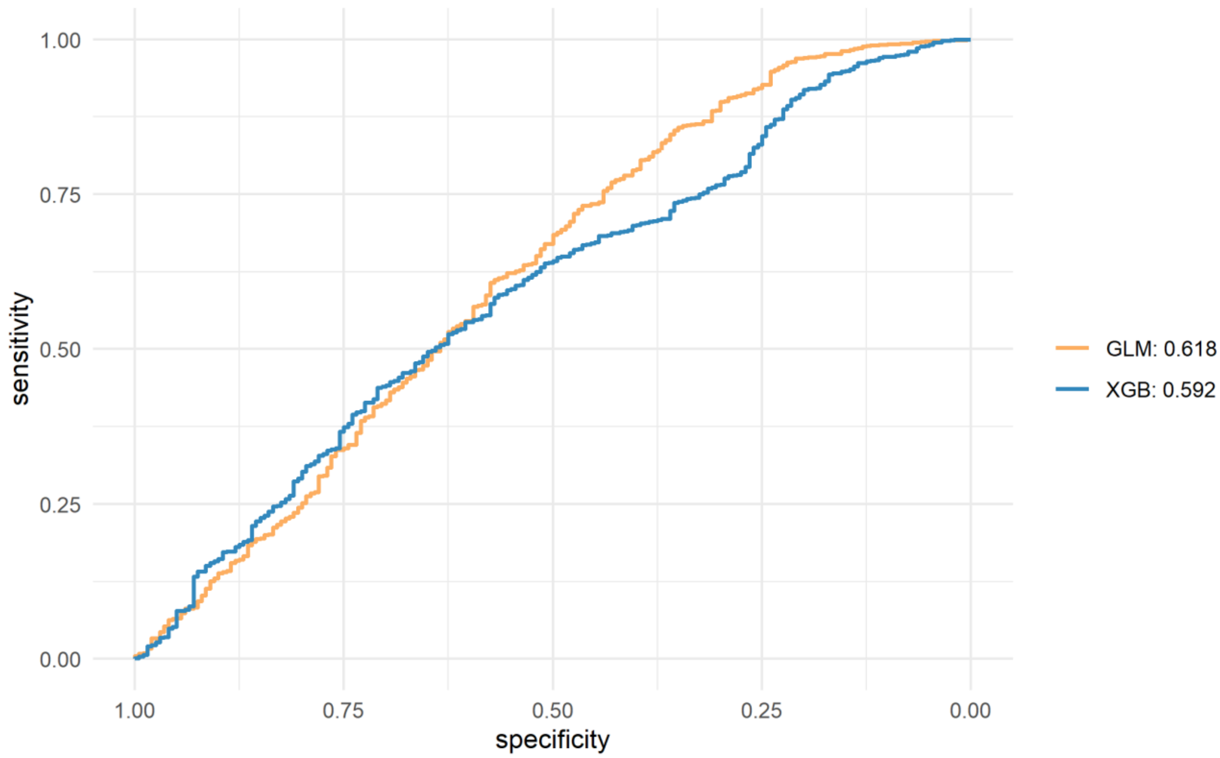

Figure 7: ROC Full Models

3.4.2 First Trimester Exposure

Next, both the XGBoost and GLM classifiers were refit using pollution exposure from only the first trimester. The XGBoost model correctly classified 41 abnormal Apgar scores and 16, 424 normal Apgar scores (Table 8). The logistic regression correctly classified 101 abnormal Apgar scores and 12,269 normal Apgar scores (Table 9).

Table 8: Confusion Matrix: XGBoost First Trimester

		Reference	
		Abnormal	*Normal*
Prediction	*Abnormal*	41	1,711
	Normal	159	16,424

Table 9: Confusion Matrix: GLM First Trimester

		Reference	
		Abnormal	*Normal*
Prediction	*Abnormal*	101	5,866
	Normal	99	12,269

Additional measures of model performance can be seen in Table 10. Neither model's predictive accuracy was significantly greater than the no information rate (Table 10).

Table 10: Performance Statistics

	Sensitivity	Specificity	PPV	NPV	Accuracy	NIR	P(Acc > NIR)
XGBoost	0.205	0.906	0.023	0.990	0.898	0.989	1
GLM	0.505	0.676	0.017	0.992	0.675	0.989	1

In this model, the logistic regression's performance was slightly worse than that of the XGBoost classifier, as shown by the ROC curves in Figure 8; the AUC of logistic regression was 0.619 and the AUC of the XGBoost classifier was 0.622.

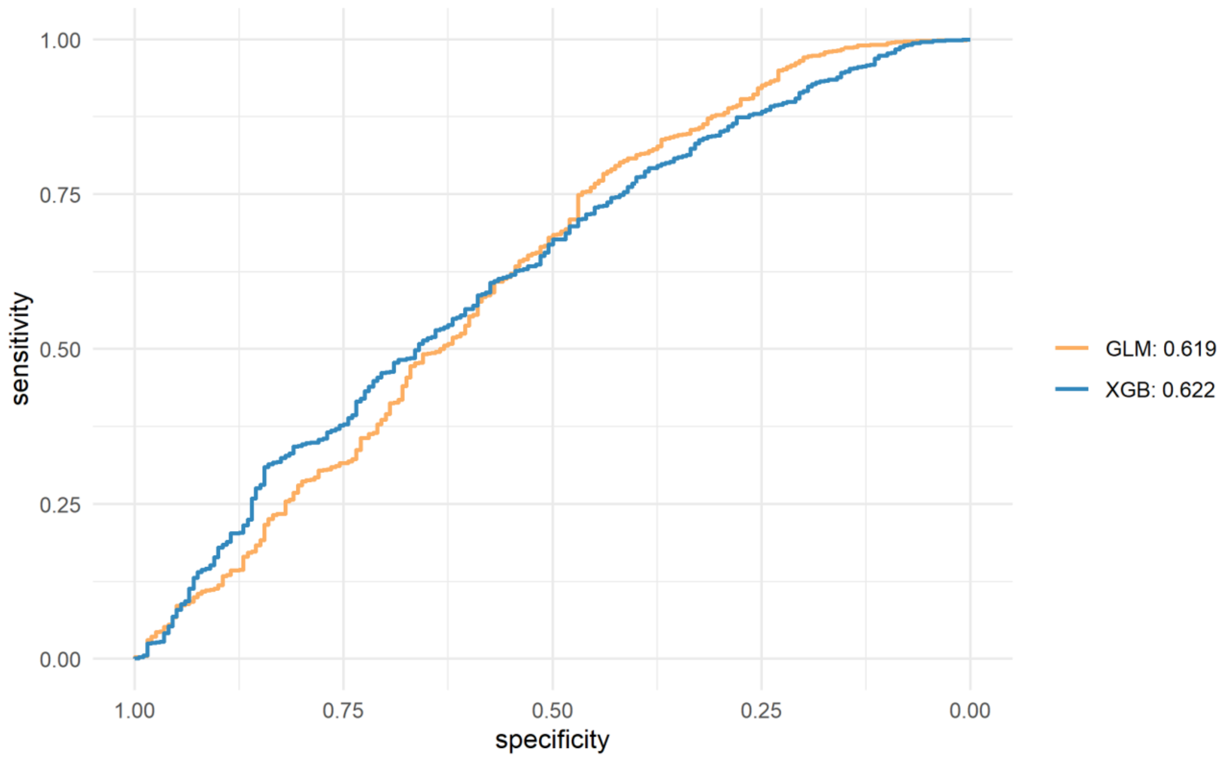

Figure 8: ROC First Trimester

3.4.3 Second Trimester Exposure

Both the XGBoost and GLM classifiers were again refit, using the pollution exposures from the second trimester only. Using this construction, the XGBoost classifier correctly predicted 47 abnormal Apgar scores and 16,347 normal Apgar scores (Table 11); the logistic regression correctly predicted 103 abnormal Apgar scores and 12,297 normal Apgar scores (Table 12).

Table 11: Confusion Matrix - XGBoost Second Trimester

		Reference	
		Abnormal	*Normal*
Prediction	*Abnormal*	47	1,788
	Normal	153	16,347

Table 12: Confusion Matrix - GLM Second Trimester

		Reference	
		Abnormal	*Normal*
Prediction	*Abnormal*	103	5,838
	Normal	97	12,297

Further metrics of model performance can be seen in Table 13. Once again, neither model had an accuracy that was significantly greater than the no information rate (Table 13).

Table 13: Performance Statistics

	Sensitivity	Specificity	PPV	NPV	Accuracy	NIR	P(Acc > NIR)
XGBoost	0.225	0.901	0.026	0.991	0.894	0.989	1
GLM	0.515	0.678	0.017	0.992	0.676	0.989	1

As seen in Figure 4, the AUC of the logistic regression was 0.618, whereas the AUC of the XGBoost model was 0.627.

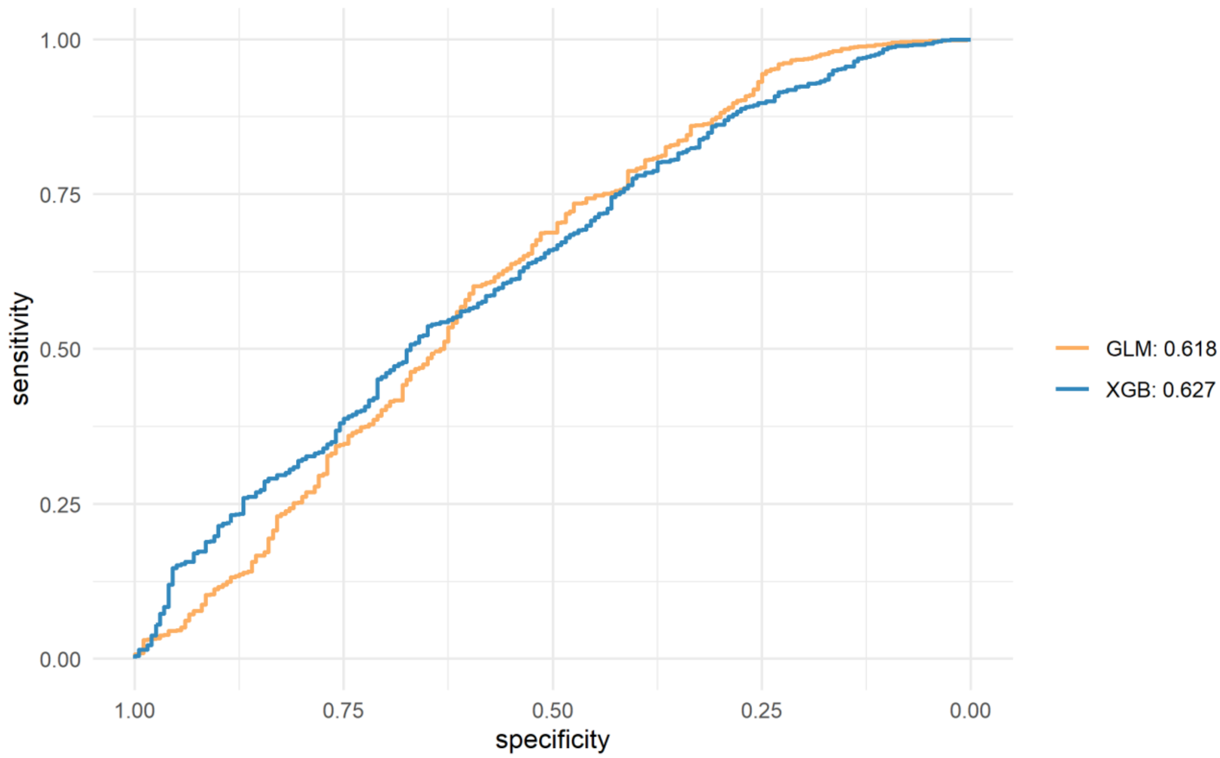

Figure 9: ROC Second Trimester

3.4.4 Third Trimester Pollution Exposure

Once again, the models were rebuilt – this time, using air pollution exposures from the third trimester only. Using these parameters, the XGBoost model correctly identified 33 abnormal Apgar scores and 16,620 normal Apgar scores (Table 14); the GLM correctly identified 97 abnormal Apgar scores and 12,334 normal Apgar scores (Table 15).

Table 14: Confusion Matrix - XGBoost Third Trimester

		Reference	
		Abnormal	*Normal*
Prediction	*Abnormal*	33	1,515
	Normal	167	16,620

Table 15: Confusion Matrix - GLM Third Trimester

		Reference	
		Abnormal	*Normal*
Prediction	*Abnormal*	103	5,838
	Normal	97	12,297

Table 16 shows additional measures of model performance. Neither model had a predictive accuracy that was significantly greater than the no information rate (Table 16).

Table 16: Performance Statistics

	Sensitivity	Specificity	PPV	NPV	Accuracy	NIR	P(Acc > NIR)
XGBoost	0.165	0.916	0.021	0.990	0.908	0.989	1
GLM	0.485	0.680	0.016	0.992	0.678	0.989	1

In this model, the logistic regression had a marginally better AUC than the XGBoost classifier. The ROC curves are shown in Figure 5 – the AUC of the logistic regression was 0.609 and the AUC of the XGBoost model was 0.565.

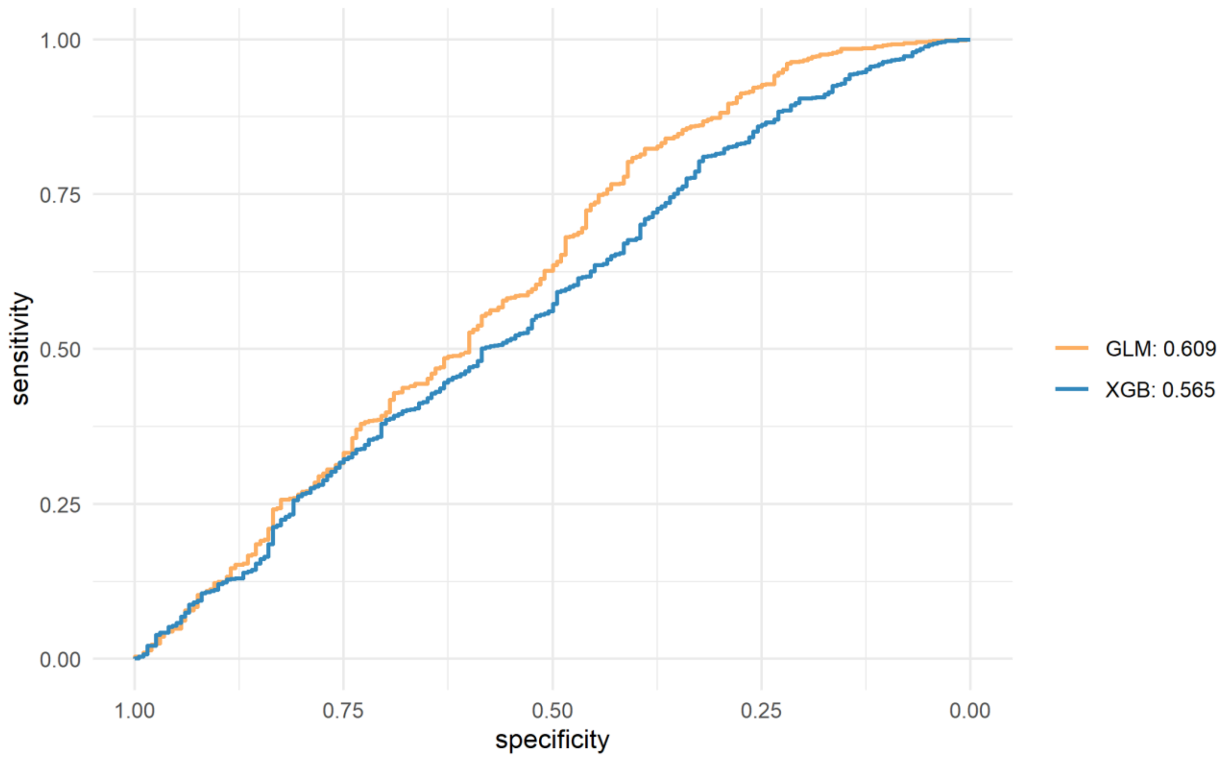

Figure 10: ROC Third Trimester

4.0 Discussion

The aims of this study were to develop a model that could accurately predict whether an Apgar score would be normal or abnormal and to assess whether maternal exposure to air pollution was important in making these predictions. Of the models fit in this study, all performed quite underwhelmingly. The overall best performing model was the second trimester XGBoost model with an AUC of 0.627. Nonetheless, this was only marginally better than the results of the other models. Furthermore, it appears that model performance did not substantially differ by trimester, indicating that air pollution exposure during one particular trimester is not more important than another; however, we cannot conclude this with much certainty, given the unreliability of our models.

The logistic regression models identified a number of statistically significant predictors. According to the full model, gestational age was by far the most significant predictor of Apgar score. Many of the air pollution variables were also found to be significant; however, some coefficients were calculated as being negative, which is the opposite of what we would expect. This can most likely be attributed to the unreliability of the model or to the possibility that some pollutants do not have a clinically significant association with Apgar score.

When fitting the XGBoost classifiers, gestational age and air pollution exposure seemed to be the most important variables in predicting Apgar score. In each of the models, gestational age was overwhelmingly the most influential feature when measuring importance by gain. Furthermore, the partial dependence plot of Apgar score and gestational age reveals that the relationship is non-linear. The probability of an abnormal Apgar score is high with a low gestational age and decreases until around 40 weeks of gestation, then begins to increase beyond

40 weeks. This finding suggests that XGBoost may be a more appropriate modeling strategy than logistic regression. While logistic regression assumes a linear relationship between predictors and the outcome, tree-based methods, including XGBoost, have the capability of capturing non-linear relationships.

The inconclusive results regarding the importance and significance of air pollution exposure in predicting Apgar score is somewhat surprising, given that several studies have found evidence supporting an association between exposure to certain air pollutants and a low Apgar score. For example, a similar study conducted in Guangzhou, China found that exposure to soil dust, a constituent of $PM_{2.5}$, significantly increased odds of an abnormal Apgar score at one minute (Wei et al., 2021). Furthermore, a study focused on South African women found that exposure to NO_x pollution, which includes nitrogen dioxide, was negatively associated with both one- and five-minute Apgar scores for infants born to mothers of a particular genotype (Naidoo, Naidoo, Ramkaran, & Chuturgoon, 2020). Further exploration is required to determine with certainty if such associations exist in Allegheny County.

4.1 Limitations

There were several limitations to this study. Most notably, the class distribution of the outcome was heavily imbalanced. A mere 1% of births had an abnormal Apgar score, with nearly the entirety of births the dataset having a normal Apgar score. In order to balance the classes, a large quantity of data in the majority class was lost, while a large quantity in the minority class was repeated. This likely introduced a considerable amount of bias into the data, resulting in poorly performing models.

Additionally, the accuracy of the air pollution exposures was undoubtedly hindered by the nature of the air quality data. The air quality data represent pollutant concentrations at specific points; our method of exposure assignment does not take into consideration any variation in concentration levels that may occur due to distance or geological factors. Furthermore, some pollutants were monitored at significantly fewer sites than others. Consequently, the distances between mothers' residences and monitors were larger, which likely affected accuracy as well.

4.2 Future Directions

Future research on the topic of Apgar score in relation to air pollution exposure can expand by addressing the limitations of this study. More sophisticated resampling techniques should be considered in order to generate synthetic data more accurately. The Synthetic Minority Oversampling Technique (SMOTE) and the Adaptive Synthetic Sampling Approach (ADASYN) are two popular oversampling techniques that generate new minority class observations by learning from existing observations. These can be used in conjunction with data-driven under-sampling techniques, such as Edited Nearest Neighbors (ENN) or Tomek Links. While these techniques are computationally intensive, they can provide superior results to random resampling.

Moreover, exploring geospatial modelling of air pollutant concentrations could prove beneficial in improving the precision of exposure calculations. A similar study also conducted in Allegheny county utilized space-time ordinary kriging (STOK) interpolation to estimate pollutant concentrations at the centroids of a grid (Lee, Roberts, Catov, Talbott, & Ritz, 2013)

This technique can also be computationally intensive depending on the size of the data being used, but could significantly improve the quality of exposure assignment.

Appendix A Air Quality Monitor Maps

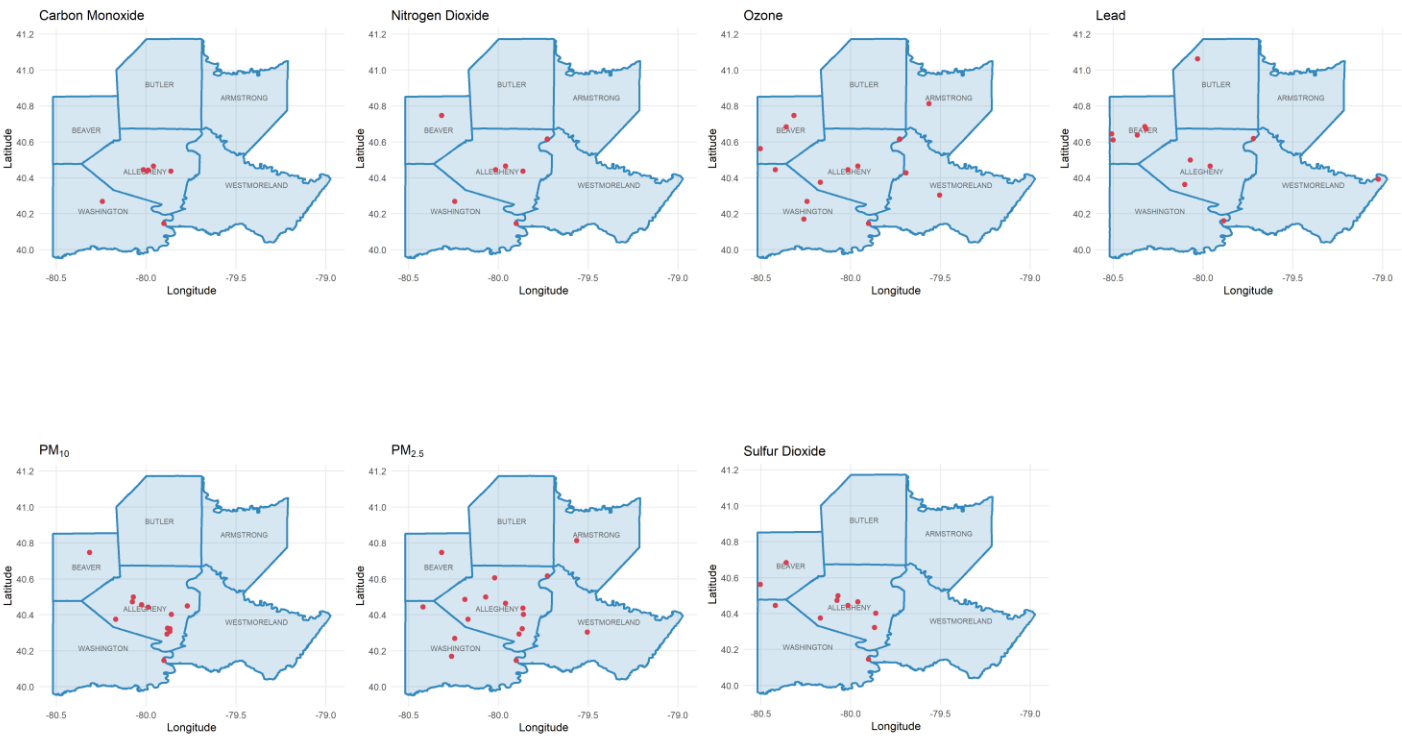

Figure 11: Air Quality Monitor Maps

Appendix B Trimester-Specific Logistic Regression

Table 9: First Trimester Logistic Regression Output

	Estimate	Standard Error	z value	Pr(>|z|) Signif.
(Intercept)	9.216	0.238	38.713	0.0000 ***
sex.M	0.328	0.021	15.751	0.0000 ***
season_of_birth.Spring	-0.048	0.034	-1.442	0.1492
season_of_birth.Summer	-0.003	0.037	-0.072	0.9427
season_of_birth.Winter	-0.096	0.036	-2.673	0.0075 **
gestational_age_weeks	-0.220	0.004	-49.391	0.0000 ***
mothage	0.004	0.002	1.746	0.0809 .
bmi_cat.Obese	-0.014	0.028	-0.495	0.6203
bmi_cat.Overweight	-0.004	0.026	-0.140	0.8890
bmi_cat.Underweight	-0.811	0.072	-11.196	0.0000 ***
gestational_diabetes.Yes	0.107	0.044	2.436	0.0148 *
maternal_race.Black.or.African.American	0.526	0.076	6.947	0.0000 ***
maternal_race.White	0.216	0.064	3.383	0.0007 ***
maternal_ethnicity.Not.Hispanic	0.167	0.085	1.963	0.0497 *
maternal_edu_cat.Graduate.degree	-0.242	0.028	-8.578	0.0000 ***
maternal_edu_cat.High.school.or.GED	-0.024	0.036	-0.662	0.5083
maternal_edu_cat.Less.than.high.school	-0.150	0.063	-2.402	0.0163 *
maternal_edu_cat.Some.college	-0.052	0.029	-1.783	0.0746 .
smkpr	0.017	0.002	8.193	0.0000 ***
smk_total	-0.002	0.001	-1.697	0.0897 .
paternal_race.Black	-0.091	0.049	-1.885	0.0595 .
paternal_race.All.other.races	0.088	0.061	1.449	0.1474
paternal_ethnicity.Hispanic	-0.442	0.082	-5.393	0.0000 ***
co_first	-0.649	0.104	-6.224	0.0000 ***
no2_first	-0.053	0.004	-15.165	0.0000 ***
ozone_first	-25.909	2.291	-11.310	0.0000 ***
pb_first	-2.475	0.482	-5.131	0.0000 ***
pm10_first	0.002	0.003	0.687	0.4918
pm2.5_first	0.033	0.006	5.073	0.0000 ***

	Estimate	Standard Error	z value	Pr(>|z|)	Signif.
so2_first	-0.129	0.007	-17.869	0.0000	***

*Signif. codes: 0 <= '***' < 0.001 < '**' < 0.01 < '*' < 0.05 < '.' < 0.1 < '' < 1*

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 5.931e+04 on 42782 degrees of freedom

Residual deviance: 5.41e+04 on 42753 degrees of freedom

Table 10: Second Trimester Logistic Regression Output

	Estimate	Standard Error	z value	Pr(>|z|) Signif.
(Intercept)	9.545	0.247	38.700	0.0000 ***
sex.M	0.320	0.021	15.375	0.0000 ***
season_of_birth.Spring	0.084	0.043	1.955	0.0506 .
season_of_birth.Summer	0.217	0.033	6.483	0.0000 ***
season_of_birth.Winter	-0.085	0.036	-2.352	0.0187 *
gestational_age_weeks	-0.219	0.004	-48.682	0.0000 ***
mothage	0.004	0.002	1.588	0.1124
bmi_cat.Obese	-0.026	0.028	-0.937	0.3486
bmi_cat.Overweight	0.010	0.025	0.376	0.7069
bmi_cat.Underweight	-0.752	0.073	-10.365	0.0000 ***
gestational_diabetes.Yes	0.114	0.044	2.601	0.0093 **
maternal_race.Black.or.African.American	0.548	0.076	7.200	0.0000 ***
maternal_race.White	0.216	0.064	3.372	0.0007 ***
maternal_ethnicity.Not.Hispanic	0.093	0.086	1.085	0.2781
maternal_edu_cat.Graduate.degree	-0.262	0.028	-9.299	0.0000 ***
maternal_edu_cat.High.school.or.GED	-0.026	0.036	-0.712	0.4766
maternal_edu_cat.Less.than.high.school	-0.209	0.063	-3.343	0.0008 ***
maternal_edu_cat.Some.college	-0.051	0.029	-1.741	0.0818 .
smkpr	0.018	0.002	8.395	0.0000 ***
smk_total	-0.003	0.001	-2.453	0.0142 *
paternal_race.Black	-0.038	0.049	-0.786	0.4318
paternal_race.All.other.races	0.084	0.061	1.378	0.1683
paternal_ethnicity.Hispanic	-0.455	0.083	-5.506	0.0000 ***
co_second	-0.954	0.103	-9.302	0.0000 ***
no2_second	-0.046	0.004	-12.702	0.0000 ***
ozone_second	-26.535	2.330	-11.389	0.0000 ***
pb_second	-0.384	0.442	-0.868	0.3851
pm10_second	-0.025	0.003	-8.416	0.0000 ***
pm2.5_second	0.035	0.006	5.474	0.0000 ***
so2_second	-0.098	0.007	-13.383	0.0000 ***

*Signif. codes: 0 <= '***' < 0.001 < '**' < 0.01 < '*' < 0.05 < '.' < 0.1 < '' < 1*

(Dispersion parameter for binomial family taken to be 1)

	Estimate	Standard Error	z value	Pr(>|z|) Signif.

Null deviance: 5.931e+04 on 42782 degrees of freedom

Residual deviance: 5.414e+04 on 42753 degrees of freedom

Table 11: Third Trimester Logistic Regression Output

	Estimate	Standard Error	z value	Pr(>|z|) Signif.
(Intercept)	-8.596	0.237	-36.282	0.0000 ***
sex.M	-0.325	0.021	-15.623	0.0000 ***
season_of_birth.Spring	-0.287	0.033	-8.821	0.0000 ***
season_of_birth.Summer	-0.179	0.035	-5.104	0.0000 ***
season_of_birth.Winter	-0.154	0.034	-4.503	0.0000 ***
gestational_age_weeks	0.219	0.004	48.983	0.0000 ***
mothage	-0.003	0.002	-1.296	0.1951
bmi_cat.Obese	0.036	0.028	1.316	0.1881
bmi_cat.Overweight	-0.035	0.025	-1.364	0.1724
bmi_cat.Underweight	0.855	0.072	11.806	0.0000 ***
gestational_diabetes.Yes	-0.100	0.044	-2.274	0.0230 *
maternal_race.Black.or.African.American	-0.594	0.075	-7.895	0.0000 ***
maternal_race.White	-0.242	0.064	-3.809	0.0001 ***
maternal_ethnicity.Not.Hispanic	-0.179	0.085	-2.107	0.0351 *
maternal_edu_cat.Graduate.degree	0.266	0.028	9.479	0.0000 ***
maternal_edu_cat.High.school.or.GED	0.030	0.036	0.814	0.4155
maternal_edu_cat.Less.than.high.school	0.188	0.063	2.995	0.0027 **
maternal_edu_cat.Some.college	0.045	0.029	1.538	0.1240
smkpr	-0.018	0.002	-8.324	0.0000 ***
smk_total	0.003	0.001	2.349	0.0188 *
paternal_race.Black	0.059	0.048	1.221	0.2220
paternal_race.All.other.races	-0.107	0.061	-1.767	0.0772 .
paternal_ethnicity.Hispanic	0.426	0.082	5.227	0.0000 ***
co_third	0.522	0.101	5.170	0.0000 ***
no2_third	0.034	0.004	9.748	0.0000 ***
ozone_third	5.502	2.423	2.271	0.0232 *
pb_third	-1.060	0.422	-2.511	0.0120 *
pm10_third	0.009	0.003	2.905	0.0037 **
pm2.5_third	0.001	0.007	0.094	0.9251
so2_third	0.113	0.008	14.656	0.0000 ***

*Signif. codes: 0 <= '***' < 0.001 < '**' < 0.01 < '*' < 0.05 < '.' < 0.1 < '' < 1*

(Dispersion parameter for binomial family taken to be 1)
Null deviance: 5.931e+04 on 42782 degrees of freedom
Residual deviance: 5.428e+04 on 42753 degrees of freedom

Appendix C R Code

```
library(tidyverse)
library(sf)
library(RMariaDB)
library(lubridate)


birth_data_allegheny_all_vars <- read.csv("~/Nina_cleaned_birth_data.csv")
%>%
  filter(final_momrescounty_with_pgh == 'Allegheny (City of Pittsburgh)' |
           final_momrescounty_with_pgh == 'Allegheny (excl. City of
Pittsburgh)')


# apply exclusion criteria
birth_data_filtered <- birth_data_allegheny_all_vars %>%
    filter(sex != 'U' &
            multiple_birth == 0 &
            bweight_less_than_500_g == 0 &
            missing_gestational_age == 0 &
            missing_apgar5 == 0 &
            mothage <= 45 &
            bmi_cat != 'Unknown' &
            maternal_race != 'Unknown or refused' &
            maternal_ethnicity != 'Unknown' &
            maternal_edu_cat != 'Unknown' &
            smkpr < 99 &
            smkftm < 99 &
            smkstm < 99 &
            smkltm < 99 &
            gestational_age_weeks > 22)   %>%
    mutate(smk_total = smkftm + smkstm + smkltm) %>%
    select(birth_id, mother_id, sex, child_dob, season_of_birth,
gestational_age_weeks, mothage, bmi_cat, gestational_diabetes, maternal_race,
maternal_ethnicity,maternal_edu_cat, final_lat, final_long, smkpr, smk_total)


# extract additional variables
con <- dbConnect(RMariaDB::MariaDB(),
                 default.file  = "C:/Users/testuser/.my.ini",
                 group = "fracking-group")
sql_statement <- "select
                      birth_id,
                      fathrace,
                      fathhisp,
                      apgars5,
                      apgars10
                  from
                    birth_data_Combined"
sql_vars <- dbGetQuery(conn = con, statement = sql_statement)
```

```
# collapse race and ethnicity
sql_vars_clean <- sql_vars %>%
    mutate(birth_id = as.double(birth_id),
            paternal_race = fct_collapse(factor(sql_vars$fathrace), White =
'1', Black = '2','All other races' = c('3', '4', '5', '6', '7', '8', '9',
'10', '11', '12', '13','14', '15'), 'unknown/refused' = c('16', '17')),
            paternal_ethnicity = fct_collapse(factor(sql_vars$fathhisp),'Not
Hispanic' = '1', 'Hispanic' = c('2', '3', '4', '5'),'Unknown' = '9')) %>%
    select(birth_id,paternal_race, paternal_ethnicity, apgars5, apgars10)


# join larger df with additional vars
# drop unknown race, ethnicity, apgar
# categorize apgar
birth_data_final <- left_join(birth_data_filtered, sql_vars_clean, by =
'birth_id') %>%
    filter(paternal_race != 'unknown/refused' & paternal_ethnicity !=
'Unknown') %>%
    mutate(apgar5_cat = cut(apgars5, breaks = c(0, 7, 11, 100),
include.lowest = T, right = F, labels = c('abnormal', 'normal', 'missing')),
apgar10_cat = cut(apgars10, breaks = c(0, 7, 11, 99, 100), include.lowest =
T, right = F, labels = c('abnormal', 'normal', 'not applicable', 'missing')),
    child_dob = as.Date(child_dob, '%Y-%m-%d')) %>%
    mutate(first_tri_date = child_dob - gestational_age_weeks*7, # estimate
date of conception
            second_tri_date = first_tri_date + 7*13, # estimate beginning of
second trimester
            third_tri_date = first_tri_date + 7*26, # estimate beginning of
third trimester
            first_tri = interval(first_tri_date, second_tri_date),
            second_tri = interval(second_tri_date, third_tri_date),
            third_tri = interval(third_tri_date, child_dob)) %>%
    select(-c(apgars5, apgars10))


save(birth_data_final, file = 'capstone/birth_data_final.RData')

library(tidyverse)
library(sf)
library(lubridate)


load('~/capstone/birth_data_final.RData')
load("~/capstone/air quality/load aq data.RData")


# function to find active monitors
get_monitors <- function(births, tri, pollutant) {
  active_period <- pollutant %>%
    group_by(site.num) %>%
    summarize(start = min(date.local), end = max(date.local), .groups =
'drop')

  if(tri == 'first') {
    x <- apply(births, 1, function(x){x['first_tri_date'] >=
active_period$start & x['second_tri_date'] <= active_period$end})
```

```
  }
  if(tri == 'second') {
    x <- apply(births, 1, function(x){x['second_tri_date'] >=
active_period$start & x['third_tri_date'] <= active_period$end})
  }
  if(tri == 'third') {
    x <- apply(births, 1, function(x){x['third_tri_date'] >=
active_period$start & x['child_dob'] <= active_period$end})
  }

  monitors <- apply(x, 2, function(x){filter(active_period, x)$site.num})
  return(monitors)
}



# identify nearest monitor
nearest_monitor <- function(births, pollutant, monitors) {

  # create monitor sf object
  monitor_locations <- pollutant %>%
    group_by(site.num) %>%
    summarize(lat = unique(latitude), long = unique(longitude), .groups =
'drop') %>%
    st_as_sf(coords = c("long", "lat"), crs = 'WGS84', agr = "constant")

  # create birth sf object
  mom_res <- births %>%
    select(birth_id, final_lat, final_long) %>%
    st_as_sf(coords = c("final_long", "final_lat"), crs = 'NAD83', agr =
"constant") %>%
    st_transform(crs = 'WGS84')

  # for each birth, create a list of monitors that are active during
gestation period
  monitor_options  <- lapply(monitors, function(x){filter(monitor_locations,
monitor_locations$site.num %in% x)})

  # match birth to closest monitor
  f <- function(i) {
    st_join(mom_res[i,], monitor_options[[i]], join = st_nearest_feature) %>%
      as.data.frame() %>%
      select(birth_id, site.num)
  }

  results <- sapply(1:nrow(mom_res), f) %>%
                  t() %>%
                  as.data.frame() %>%
                  mutate(birth_id = sapply(birth_id, unlist),
                         site.num = sapply(site.num, unlist))

  return(results)
}


# get co active monitors
co_active_monitors_first <- get_monitors(birth_data_final, 'first', co)
```

```
co_active_monitors_second <- get_monitors(birth_data_final, 'second', co)
co_active_monitors_third <- get_monitors(birth_data_final, 'third', co)

#get no2 active monitors
no2_active_monitors_first <- get_monitors(birth_data_final, 'first', no2)
no2_active_monitors_second <- get_monitors(birth_data_final, 'second', no2)
no2_active_monitors_third <- get_monitors(birth_data_final, 'third', no2)


#get ozone active monitors
ozone_active_monitors_first <- get_monitors(birth_data_final, 'first', ozone)
ozone_active_monitors_second <- get_monitors(birth_data_final, 'second',
ozone)
ozone_active_monitors_third <- get_monitors(birth_data_final, 'third', ozone)


#get pb active monitors
pb_active_monitors_first <- get_monitors(birth_data_final, 'first', pb)
pb_active_monitors_second <- get_monitors(birth_data_final, 'second', pb)
pb_active_monitors_third <- get_monitors(birth_data_final, 'third', pb)


#get pm10 active monitors
pm10_active_monitors_first <- get_monitors(birth_data_final, 'first', pm10)
pm10_active_monitors_second <- get_monitors(birth_data_final, 'second', pm10)
pm10_active_monitors_third <- get_monitors(birth_data_final, 'third', pm10)


# get pm 2.5 active monitors
pm2.5_active_monitors_first <- get_monitors(birth_data_final, 'first', pm2.5)
pm2.5_active_monitors_second <- get_monitors(birth_data_final, 'second',
pm2.5)
pm2.5_active_monitors_third <- get_monitors(birth_data_final, 'third', pm2.5)


#get so2 active monitors
so2_active_monitors_first <- get_monitors(birth_data_final, 'first', so2)
so2_active_monitors_second <- get_monitors(birth_data_final, 'second', so2)
so2_active_monitors_third <- get_monitors(birth_data_final, 'third', so2)


save.image('~/capstone/active_monitors.RData')


# find co nearest monitors
co_nearest_monitor_first <- nearest_monitor(birth_data_final, co,
co_active_monitors_first)
co_nearest_monitor_second <- nearest_monitor(birth_data_final, co,
co_active_monitors_second)
co_nearest_monitor_third <- nearest_monitor(birth_data_final, co,
co_active_monitors_third)


# find no2 nearest monitors
no2_nearest_monitor_first <- nearest_monitor(birth_data_final, no2,
no2_active_monitors_first)
```

```
no2_nearest_monitor_second <- nearest_monitor(birth_data_final, no2,
no2_active_monitors_second)
no2_nearest_monitor_third <- nearest_monitor(birth_data_final, no2,
no2_active_monitors_third)


# find o3 nearest monitors
ozone_nearest_monitor_first <- nearest_monitor(birth_data_final, ozone,
ozone_active_monitors_first)
ozone_nearest_monitor_second <- nearest_monitor(birth_data_final, ozone,
ozone_active_monitors_second)
ozone_nearest_monitor_third <- nearest_monitor(birth_data_final, ozone,
ozone_active_monitors_third)



# find pb nearest monitors
pb_nearest_monitor_first <- nearest_monitor(birth_data_final, pb,
pb_active_monitors_first)
pb_nearest_monitor_second <- nearest_monitor(birth_data_final, pb,
pb_active_monitors_second)
pb_nearest_monitor_third <- nearest_monitor(birth_data_final, pb,
pb_active_monitors_third)


# find pm10 nearest monitors
pm10_nearest_monitor_first <- nearest_monitor(birth_data_final, pm10,
pm10_active_monitors_first)
pm10_nearest_monitor_second <- nearest_monitor(birth_data_final, pm10,
pm10_active_monitors_second)
pm10_nearest_monitor_third <- nearest_monitor(birth_data_final, pm10,
pm10_active_monitors_third)


# find pm2.5 nearest monitors
pm2.5_nearest_monitor_first <- nearest_monitor(birth_data_final, pm2.5,
pm2.5_active_monitors_first)
pm2.5_nearest_monitor_second <- nearest_monitor(birth_data_final, pm2.5,
pm2.5_active_monitors_second)
pm2.5_nearest_monitor_third <- nearest_monitor(birth_data_final, pm2.5,
pm2.5_active_monitors_third)


# find so2 nearest monitors
so2_nearest_monitor_first <- nearest_monitor(birth_data_final, so2,
so2_active_monitors_first)
so2_nearest_monitor_second <- nearest_monitor(birth_data_final, so2,
so2_active_monitors_second)
so2_nearest_monitor_third <- nearest_monitor(birth_data_final, so2,
so2_active_monitors_third)


#join birth id, gestation estimates, closest monitor
co_nearest_monitor_first$birth_id <-
as.double(co_nearest_monitor_first$birth_id)
co_nearest_monitor_second$birth_id <-
as.double(co_nearest_monitor_second$birth_id)
```

```
co_nearest_monitor_third$birth_id <-
as.double(co_nearest_monitor_third$birth_id)

no2_nearest_monitor_first$birth_id <-
as.double(no2_nearest_monitor_first$birth_id)
no2_nearest_monitor_second$birth_id <-
as.double(no2_nearest_monitor_second$birth_id)
no2_nearest_monitor_third$birth_id <-
as.double(no2_nearest_monitor_third$birth_id)

ozone_nearest_monitor_first$birth_id <-
as.double(ozone_nearest_monitor_first$birth_id)
ozone_nearest_monitor_second$birth_id <-
as.double(ozone_nearest_monitor_second$birth_id)
ozone_nearest_monitor_third$birth_id <-
as.double(ozone_nearest_monitor_third$birth_id)

pb_nearest_monitor_first$birth_id <-
as.double(pb_nearest_monitor_first$birth_id)
pb_nearest_monitor_second$birth_id <-
as.double(pb_nearest_monitor_second$birth_id)
pb_nearest_monitor_third$birth_id <-
as.double(pb_nearest_monitor_third$birth_id)


pm10_nearest_monitor_first$birth_id <-
as.double(pm10_nearest_monitor_first$birth_id)
pm10_nearest_monitor_second$birth_id <-
as.double(pm10_nearest_monitor_second$birth_id)
pm10_nearest_monitor_third$birth_id <-
as.double(pm10_nearest_monitor_third$birth_id)


pm2.5_nearest_monitor_first$birth_id <-
as.double(pm2.5_nearest_monitor_first$birth_id)
pm2.5_nearest_monitor_second$birth_id <-
as.double(pm2.5_nearest_monitor_second$birth_id)
pm2.5_nearest_monitor_third$birth_id <-
as.double(pm2.5_nearest_monitor_third$birth_id)


so2_nearest_monitor_first$birth_id <-
as.double(so2_nearest_monitor_first$birth_id)
so2_nearest_monitor_second$birth_id <-
as.double(so2_nearest_monitor_second$birth_id)
so2_nearest_monitor_third$birth_id <-
as.double(so2_nearest_monitor_third$birth_id)


first_trimester_co <- birth_data_final %>%
                      select(birth_id, first_tri_date, second_tri_date) %>%
                      left_join(co_nearest_monitor_first)
second_trimester_co <- birth_data_final %>%
                        select(birth_id, second_tri_date, third_tri_date) %>%
                        left_join(co_nearest_monitor_second)
third_trimester_co <- birth_data_final %>%
                        select(birth_id, third_tri_date, child_dob) %>%
```

```
                          left_join(co_nearest_monitor_third)


first_trimester_no2 <- birth_data_final %>%
                            select(birth_id, first_tri_date, second_tri_date)
%>%
                            left_join(no2_nearest_monitor_first)
second_trimester_no2 <- birth_data_final %>%
                            select(birth_id, second_tri_date, third_tri_date)
%>%
                            left_join(no2_nearest_monitor_second)
third_trimester_no2 <- birth_data_final %>%
                          select(birth_id, third_tri_date, child_dob) %>%
                          left_join(no2_nearest_monitor_third)


first_trimester_ozone <- birth_data_final %>%
                            select(birth_id, first_tri_date, second_tri_date)
%>%
                            left_join(ozone_nearest_monitor_first)
second_trimester_ozone <- birth_data_final %>%
                            select(birth_id, second_tri_date, third_tri_date)
%>%
                            left_join(ozone_nearest_monitor_second)
third_trimester_ozone <- birth_data_final %>%
                          select(birth_id, third_tri_date, child_dob) %>%
                          left_join(ozone_nearest_monitor_third)



first_trimester_pb <- birth_data_final %>%
                        select(birth_id, first_tri_date, second_tri_date) %>%
                        left_join(pb_nearest_monitor_first)
second_trimester_pb <- birth_data_final %>%
                          select(birth_id, second_tri_date, third_tri_date)
%>%
                          left_join(pb_nearest_monitor_second)
third_trimester_pb <- birth_data_final %>%
                          select(birth_id, third_tri_date, child_dob) %>%
                          left_join(pb_nearest_monitor_third)



first_trimester_pm10 <- birth_data_final %>%
  select(birth_id, first_tri_date, second_tri_date) %>%
  left_join(pm10_nearest_monitor_first)
second_trimester_pm10 <- birth_data_final %>%
  select(birth_id, second_tri_date, third_tri_date) %>%
  left_join(pm10_nearest_monitor_second)
third_trimester_pm10 <- birth_data_final %>%
  select(birth_id, third_tri_date, child_dob) %>%
  left_join(pm10_nearest_monitor_third)


first_trimester_pm2.5 <- birth_data_final %>%
  select(birth_id, first_tri_date, second_tri_date) %>%
```

```
  left_join(pm2.5_nearest_monitor_first)
second_trimester_pm2.5 <- birth_data_final %>%
  select(birth_id, second_tri_date, third_tri_date) %>%
  left_join(pm2.5_nearest_monitor_second)
third_trimester_pm2.5 <- birth_data_final %>%
  select(birth_id, third_tri_date, child_dob) %>%
  left_join(pm2.5_nearest_monitor_third)



first_trimester_so2 <- birth_data_final %>%
  select(birth_id, first_tri_date, second_tri_date) %>%
  left_join(so2_nearest_monitor_first)
second_trimester_so2 <- birth_data_final %>%
  select(birth_id, second_tri_date, third_tri_date) %>%
  left_join(so2_nearest_monitor_second)
third_trimester_so2 <- birth_data_final %>%
  select(birth_id, third_tri_date, child_dob) %>%
  left_join(so2_nearest_monitor_third)



# set all negative concentrations to 0
co[co$arithmetic.mean < 0,]$arithmetic.mean <- 0
no2[no2$arithmetic.mean < 0,]$arithmetic.mean <- 0
ozone[ozone$arithmetic.mean < 0,]$arithmetic.mean <- 0
pb[pb$arithmetic.mean < 0,]$arithmetic.mean <- 0
pm10[pm10$arithmetic.mean < 0,]$arithmetic.mean <- 0
pm2.5[pm2.5$arithmetic.mean < 0,]$arithmetic.mean <- 0
so2[so2$arithmetic.mean < 0,]$arithmetic.mean <- 0


# find average concentrations and add to data frame
co_concentration_first <- function(i){
  data <- filter(co, site.num == first_trimester_co[i,'site.num'] &
                 date.local >= first_trimester_co[i,'first_tri_date'] &
                 date.local <= first_trimester_co[i,'second_tri_date'])
  return(mean(data$arithmetic.mean))
}
co_concentration_second <- function(i){
  data <- filter(co, site.num == second_trimester_co[i,'site.num'] &
                 date.local >= second_trimester_co[i,'second_tri_date'] &
                 date.local <= second_trimester_co[i,'third_tri_date'])
  return(mean(data$arithmetic.mean))
}
co_concentration_third <- function(i){
  data <- filter(co, site.num == third_trimester_co[i,'site.num'] &
                 date.local >= third_trimester_co[i,'third_tri_date'] &
                 date.local <= third_trimester_co[i,'child_dob'])
  return(mean(data$arithmetic.mean))
}

birth_data_final$co_first <- sapply(1:nrow(birth_data_final),
co_concentration_first)
birth_data_final$co_second <- sapply(1:nrow(birth_data_final),
co_concentration_second)
```

```
birth_data_final$co_third <- sapply(1:nrow(birth_data_final),
co_concentration_third)



no2_concentration_first <- function(i){
    data <- filter(no2, site.num == first_trimester_no2[i,'site.num'] &
                      date.local >= first_trimester_no2[i,'first_tri_date'] &
                      date.local <= first_trimester_no2[i,'second_tri_date'])
    return(mean(data$arithmetic.mean))
}
no2_concentration_second <- function(i) {
  data <- filter(no2, site.num == second_trimester_no2[i,'site.num'] &
                   date.local >= second_trimester_no2[i,'second_tri_date'] &
                   date.local <= second_trimester_no2[i,'third_tri_date'])
  return(mean(data$arithmetic.mean))
}
no2_concentration_third <- function(i) {
  data <- filter(no2, site.num == third_trimester_no2[i,'site.num'] &
                   date.local >= third_trimester_no2[i,'third_tri_date'] &
                   date.local <= third_trimester_no2[i,'child_dob'])
  return(mean(data$arithmetic.mean))
}

birth_data_final$no2_first <- sapply(1:nrow(birth_data_final),
no2_concentration_first)
birth_data_final$no2_second <- sapply(1:nrow(birth_data_final),
no2_concentration_second)
birth_data_final$no2_third <- sapply(1:nrow(birth_data_final),
no2_concentration_third)


ozone_concentration_first <- function(i){
  data <- filter(ozone, site.num == first_trimester_ozone[i,'site.num'] &
                   date.local >= first_trimester_ozone[i,'first_tri_date'] &
                   date.local <= first_trimester_ozone[i,'second_tri_date'])
  return(mean(data$arithmetic.mean))
}
ozone_concentration_second <- function(i) {
  data <- filter(ozone, site.num == second_trimester_ozone[i,'site.num'] &
                   date.local >= second_trimester_ozone[i,'second_tri_date']
&
                   date.local <= second_trimester_ozone[i,'third_tri_date'])
  return(mean(data$arithmetic.mean))
}
ozone_concentration_third <- function(i) {
  data <- filter(ozone, site.num == third_trimester_ozone[i,'site.num'] &
                   date.local >= third_trimester_ozone[i,'third_tri_date'] &
                   date.local <= third_trimester_ozone[i,'child_dob'])
  return(mean(data$arithmetic.mean))
}

birth_data_final$ozone_first <- sapply(1:nrow(birth_data_final),
ozone_concentration_first)
birth_data_final$ozone_second <- sapply(1:nrow(birth_data_final),
ozone_concentration_second)
```

```
birth_data_final$ozone_third <- sapply(1:nrow(birth_data_final),
ozone_concentration_third)



pb_concentration_first <- function(i){
  data <- filter(pb, site.num == first_trimester_pb[i,'site.num'] &
                 date.local >= first_trimester_pb[i,'first_tri_date'] &
                 date.local <= first_trimester_pb[i,'second_tri_date'])
  return(mean(data$arithmetic.mean))
}
pb_concentration_second <- function(i) {
  data <- filter(pb, site.num == second_trimester_pb[i,'site.num'] &
                 date.local >= second_trimester_pb[i,'second_tri_date'] &
                 date.local <= second_trimester_pb[i,'third_tri_date'])
  return(mean(data$arithmetic.mean))
}
pb_concentration_third <- function(i) {
  data <- filter(pb, site.num == third_trimester_pb[i,'site.num'] &
                 date.local >= third_trimester_pb[i,'third_tri_date'] &
                 date.local <= third_trimester_pb[i,'child_dob'])
  return(mean(data$arithmetic.mean))
}

birth_data_final$pb_first <- sapply(1:nrow(birth_data_final),
pb_concentration_first)
birth_data_final$pb_second <- sapply(1:nrow(birth_data_final),
pb_concentration_second)
birth_data_final$pb_third <- sapply(1:nrow(birth_data_final),
pb_concentration_third)



pm10_concentration_first <- function(i){
  data <- filter(pm10, site.num == first_trimester_pm10[i,'site.num'] &
                 date.local >= first_trimester_pm10[i,'first_tri_date'] &
                 date.local <= first_trimester_pm10[i,'second_tri_date'])
  return(mean(data$arithmetic.mean))
}
pm10_concentration_second <- function(i) {
  data <- filter(pm10, site.num == second_trimester_pm10[i,'site.num'] &
                 date.local >= second_trimester_pm10[i,'second_tri_date'] &
                 date.local <= second_trimester_pm10[i,'third_tri_date'])
  return(mean(data$arithmetic.mean))
}
pm10_concentration_third <- function(i) {
  data <- filter(pm10, site.num == third_trimester_pm10[i,'site.num'] &
                 date.local >= third_trimester_pm10[i,'third_tri_date'] &
                 date.local <= third_trimester_pm10[i,'child_dob'])
  return(mean(data$arithmetic.mean))
}

birth_data_final$pm10_first <- sapply(1:nrow(birth_data_final),
pm10_concentration_first)
birth_data_final$pm10_second <- sapply(1:nrow(birth_data_final),
pm10_concentration_second)
```

```
birth_data_final$pm10_third <- sapply(1:nrow(birth_data_final),
pm10_concentration_third)



pm2.5_concentration_first <- function(i){
  data <- filter(pm2.5, site.num == first_trimester_pm2.5[i,'site.num'] &
                 date.local >= first_trimester_pm2.5[i,'first_tri_date'] &
                 date.local <= first_trimester_pm2.5[i,'second_tri_date'])
  return(mean(data$arithmetic.mean))
}
pm2.5_concentration_second <- function(i) {
  data <- filter(pm2.5, site.num == second_trimester_pm2.5[i,'site.num'] &
                 date.local >= second_trimester_pm2.5[i,'second_tri_date']
&
                 date.local <= second_trimester_pm2.5[i,'third_tri_date'])
  return(mean(data$arithmetic.mean))
}
pm2.5_concentration_third <- function(i) {
  data <- filter(pm2.5, site.num == third_trimester_pm2.5[i,'site.num'] &
                 date.local >= third_trimester_pm2.5[i,'third_tri_date'] &
                 date.local <= third_trimester_pm2.5[i,'child_dob'])
  return(mean(data$arithmetic.mean))
}

birth_data_final$pm2.5_first <- sapply(1:nrow(birth_data_final),
pm2.5_concentration_first)
birth_data_final$pm2.5_second <- sapply(1:nrow(birth_data_final),
pm2.5_concentration_second)
birth_data_final$pm2.5_third <- sapply(1:nrow(birth_data_final),
pm2.5_concentration_third)



so2_concentration_first <- function(i){
  data <- filter(so2, site.num == first_trimester_so2[i,'site.num'] &
                 date.local >= first_trimester_so2[i,'first_tri_date'] &
                 date.local <= first_trimester_so2[i,'second_tri_date'])
  return(mean(data$arithmetic.mean))
}
so2_concentration_second <- function(i) {
  data <- filter(so2, site.num == second_trimester_so2[i,'site.num'] &
                 date.local >= second_trimester_so2[i,'second_tri_date'] &
                 date.local <= second_trimester_so2[i,'third_tri_date'])
  return(mean(data$arithmetic.mean))
}
so2_concentration_third <- function(i) {
  data <- filter(so2, site.num == third_trimester_so2[i,'site.num'] &
                 date.local >= third_trimester_so2[i,'third_tri_date'] &
                 date.local <= third_trimester_so2[i,'child_dob'])
  return(mean(data$arithmetic.mean))
}

birth_data_final$so2_first <- sapply(1:nrow(birth_data_final),
so2_concentration_first)
birth_data_final$so2_second <- sapply(1:nrow(birth_data_final),
so2_concentration_second)
```

```
birth_data_final$so2_third <- sapply(1:nrow(birth_data_final),
so2_concentration_third)

save.image('~/capstone/exposure.RData')
save(birth_data_final, file = '~/capstone/birth_data_final.RData')

library(tidyverse)
library(caret)
library(pROC)
library(smotefamily)
library(xgboost)
library(ROSE)
library(svMisc)


load('~/capstone/birth_data_final.RData')

# select variables of interest and convert categorical vars to factors
data <- birth_data_final %>%
  select(-birth_id, -mother_id, -child_dob, -final_lat, -final_long, -
first_tri_date,
         -second_tri_date, -third_tri_date, -first_tri, -second_tri, -
third_tri) %>%
  na.omit() %>%
  mutate(sex = as.factor(sex),
         season_of_birth = as.factor(season_of_birth),
         bmi_cat = as.factor(bmi_cat),
         gestational_diabetes = as.factor(gestational_diabetes),
         maternal_race = as.factor(maternal_race),
         maternal_ethnicity = as.factor(maternal_ethnicity),
         maternal_edu_cat = as.factor(maternal_edu_cat),
         paternal_race = factor(paternal_race, levels = c('White', 'Black',
'All other races')),
         paternal_ethnicity = factor(paternal_ethnicity, levels = c("Not
Hispanic", "Hispanic")),
         apgar5_cat = factor(apgar5_cat, levels = c('abnormal', 'normal')),
         apgar10_cat = as.factor(apgar10_cat))

apgar5_data <- data %>% select(-apgar10_cat)


#generate dummy variables
dummy <- dummyVars(" ~ .", data = apgar5_data, fullRank = T)
data_d <- data.frame(predict(dummy, newdata = apgar5_data))




set.seed(1234)

#create training and testing sets
n <- nrow(data_d)
index_train <- sample(1:n,size = round(0.7*n))
index_test <- (1:n)[-index_train]

train <- data_d[index_train,]
test <- data_d[index_test,]
```

```
# resample to balance classes
rose_train <- ovun.sample(apgar5_cat.normal ~.,
                          data = train,
                          method = 'both',
                          seed = 1234)$data



# check class distribution
table(rose_train$apgar5_cat.normal)



train_x <- select(rose_train, -apgar5_cat.normal)
train_y <- rose_train$apgar5_cat.normal



# build models
ctrl_acc <- trainControl(method = "repeatedcv", number = 5, classProbs =
TRUE, savePredictions = T)

glm <- train(factor(apgar5_cat.normal,
                    levels = c(0,1),
                    labels = c('abnormal', 'normal')) ~ ., data = rose_train,
             method = "glm",
             trControl = ctrl_acc)

xgb <- train(factor(apgar5_cat.normal,
                    levels = c(0,1),
                    labels = c('abnormal', 'normal')) ~ ., data = rose_train,
             method = "xgbTree",
             trControl = ctrl_acc)


test_x <- select(test, -apgar5_cat.normal)
test_y <- test$apgar5_cat.normal
pred_glm <- predict(glm, newdata = test_x)
pred_xgb <- predict(xgb, newdata = test_x)



test_y <- factor(test_y, levels = c(0,1), labels = c('abnormal', 'normal'))

train_first <- select(rose_train, c(1:22, co_first,
                                 no2_first,
                                 ozone_first, pb_first, pm10_first,
pm2.5_first,
                                 so2_first, apgar5_cat.normal))
test_x_first <- select(test_x, c(1:22, co_first,
                                   no2_first,
                                   ozone_first, pb_first, pm10_first,
pm2.5_first, so2_first))

glm_first <- train(factor(apgar5_cat.normal,
                    levels = c(1,0),
```

```
                    labels = c('normal', 'abnormal')) ~ ., data =
train_first,
             method = "glm",
             trControl = ctrl_acc)
xgb_first <- train(factor(apgar5_cat.normal,
                          levels = c(0,1),
                          labels = c('abnormal', 'normal')) ~ ., data =
train_first,
                   method = "xgbTree",
                   trControl = ctrl_acc)


glm_pred_first <- predict(glm_first, newdata = test_x_first)
xgb_pred_first <- predict(xgb_first, newdata = test_x_first)



train_second <- select(rose_train, c(1:22, co_second,
                                    no2_second,
                                    ozone_second, pb_second, pm10_second,
pm2.5_second,
                                    so2_second, apgar5_cat.normal))
test_x_second <- select(test_x, c(1:22, co_second,
                                 no2_second,
                                 ozone_second, pb_second, pm10_second,
pm2.5_second, so2_second))


glm_second <- train(factor(apgar5_cat.normal,
                           levels = c(1,0),
                           labels = c('normal', 'abnormal')) ~ ., data =
train_second,
                   method = "glm",
                   trControl = ctrl_acc)
xgb_second <- train(factor(apgar5_cat.normal,
                          levels = c(0,1),
                          labels = c('abnormal', 'normal')) ~ ., data =
train_second,
                   method = "xgbTree",
                   trControl = ctrl_acc)


glm_pred_second <- predict(glm_second, newdata = test_x_second)
xgb_pred_second <- predict(xgb_second, newdata = test_x_second)





train_third <- select(rose_train, c(1:22, co_third,
                                     no2_third,
                                     ozone_third, pb_third, pm10_third,
pm2.5_third,
                                     so2_third, apgar5_cat.normal))
test_x_third <- select(test_x, c(1:22, co_third,
                                  no2_third,
```

```
                                  ozone_third, pb_third, pm10_third,
pm2.5_third, so2_third))


glm_third <- train(factor(apgar5_cat.normal,
                          levels = c(1,0),
                          labels = c('normal', 'abnormal')) ~ ., data =
train_third,
                    method = "glm",
                    trControl = ctrl_acc)
xgb_third <- train(factor(apgar5_cat.normal,
                           levels = c(0,1),
                           labels = c('abnormal', 'normal')) ~ ., data =
train_third,
                    method = "xgbTree",
                    trControl = ctrl_acc)



glm_pred_third <- predict(glm_third, newdata = test_x_third)
xgb_pred_third <- predict(xgb_third, newdata = test_x_third)






glm_full <- train(factor(apgar5_cat.normal,
                         levels = c(1,0),
                         labels = c('normal', 'abnormal')) ~ .,
                   data = rose_train,
                   method = "glm",
                   family = 'binomial',
                   trControl = ctrl_acc)



glm_pred_full <- predict(glm_full, newdata = test_x)


xgb_full <- train(factor(apgar5_cat.normal,
                         levels = c(0,1),
                         labels = c('abnormal', 'normal')) ~ .,
                  data = rose_train,
                  method = "xgbTree",
                  trControl = ctrl_acc)


xgb_pred_full <- predict(xgb_full, newdata = test_x)



save.image('~/capstone/models.RData')


---
title: "final model"
```

````
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

```{r}
library(tidyverse)
library(caret)
library(pROC)
library(smotefamily)
library(xgboost)
library(ROSE)
library(svMisc)
library(table1)
library(officer)
library(flextable)
library(sf)
library(gridExtra)
```



```{r}
library(RColorBrewer)
pal <- brewer.pal(n = 11, name = 'Spectral')
```



```{r}
load('~/capstone/models.RData')
```




## Resampling results
```{r}
table(train$apgar5_cat.normal)
```

```{r}
table(rose_train$apgar5_cat.normal)
```




## Summary Stats
```{r}
labels <- list(sex = 'Sex',
               season_of_birth = 'Season of Birth',
               mothage = "Mother's Age",
               bmi_cat = "BMI",
               gestational_diabetes = "Gestational Diabetes",
````
````

```
               maternal_race = 'Maternal Race',
               maternal_ethnicity = 'Maternal Ethnicity',
               paternal_race = 'Paternal Race',
               paternal_ethnicity = 'Paternal Ethnicity',
               maternal_edu_cat = 'Maternal Education',
               smkpr = 'Number of Cigarettes Smoked Prior to Pregnancy',
               smk_total = 'Number of Cigarettes Smoked During Pregnancy',
               apgar5_cat = '5-Minute Apgar Score')
table1(~ sex +
         season_of_birth +
         gestational_age_weeks +
         mothage +
         bmi_cat +
         gestational_diabetes +
         maternal_race +
         maternal_ethnicity +
         paternal_race +
         paternal_ethnicity +
         maternal_edu_cat +
         smkpr +
         smk_total +
         apgar5_cat,
       labels,
       data = apgar5_data,
       topclass="Rtable1-times")
```

```{r}
pollutant_vars <- c('co_first', 'co_second', 'co_third', 'no2_first',
'no2_second', 'no2_third','ozone_first', 'ozone_second', 'ozone_third',
'pb_first', 'pb_second', 'pb_third', 'pm10_first', 'pm10_second',
'pm10_third', 'pm2.5_first', 'pm2.5_second', 'pm2.5_third', 'so2_first',
'so2_second',
                    'so2_third')
table1(~co_first +
        co_second +
        co_third +
         no2_first +
         no2_second +
         no2_third +
         ozone_first +
         ozone_second +
         ozone_third +
         pb_first +
         pb_second +
         pb_third +
         pm10_first +
         pm10_second +
         pm10_third +
         pm2.5_first +
         pm2.5_second +
         pm2.5_third +
         so2_first +
```

````
         so2_second +
         so2_third,
       data = apgar5_data,
       topclass="Rtable1-times")
```


## Monitor Maps
```{r}
load('~/capstone/aq boundaries.RData')
load('air quality/load aq data.RData')
```


### Carbon Monoxide

```{r}
co_lat <- unique(co$latitude)
co_long <- unique(co$longitude)
```

```{r}
co_monitors <- ggplot() +
  geom_sf(aes(geometry = all_counties$geometry),
          color = pal[10],
          fill = pal[10],
          alpha = .2,
          size = 1) +
  geom_sf_text(aes(geometry = all_counties$geometry,
                   label = toupper(all_counties$NAME)),
               size = 2.5,
               color = 'grey40') +
  geom_point(aes(x = co_long, y = co_lat),
             color = pal[2],
             size = 2) +
  theme_minimal() +
  xlab('Longitude') +
  ylab('Latitude') +
  ggtitle('Carbon Monoxide')

#ggsave('~/capstone/co monitors.png')
```


### Nitrogen Dioxide
```{r}
no2_lat <- unique(no2$latitude)
no2_long <- unique(no2$longitude)
```

```{r}
no2_monitors <- ggplot() +
  geom_sf(aes(geometry = all_counties$geometry),
          color = pal[10],
          fill = pal[10],
          alpha = .2,
          size = 1) +
````

````
  geom_sf_text(aes(geometry = all_counties$geometry,
                   label = toupper(all_counties$NAME)),
               size = 2.5,
               color = 'grey40') +
  geom_point(aes(x = no2_long, y = no2_lat),
             color = pal[2],
             size = 2) +
  theme_minimal() +
  xlab('Longitude') +
  ylab('Latitude') +
  ggtitle('Nitrogen Dioxide')

#ggsave('~/capstone/co monitors.png')
```


### Ozone
```{r}
o3_lat <- unique(ozone$latitude)
o3_long <- unique(ozone$longitude)
```

```{r}
ozone_monitors <- ggplot() +
  geom_sf(aes(geometry = all_counties$geometry),
          color = pal[10],
          fill = pal[10],
          alpha = .2,
          size = 1) +
  geom_sf_text(aes(geometry = all_counties$geometry,
                   label = toupper(all_counties$NAME)),
               size = 2.5,
               color = 'grey40') +
  geom_point(aes(x = o3_long, y = o3_lat),
             color = pal[2],
             size = 2) +
  theme_minimal() +
  xlab('Longitude') +
  ylab('Latitude') +
  ggtitle('Ozone')

#ggsave('~/capstone/co monitors.png')
```


### Lead
```{r}
pb_lat <- unique(pb$latitude)
pb_long <- unique(pb$longitude)
```

```{r}
pb_monitors <- ggplot() +
  geom_sf(aes(geometry = all_counties$geometry),
          color = pal[10],
          fill = pal[10],
          alpha = .2,
````

````
          size = 1) +
  geom_sf_text(aes(geometry = all_counties$geometry,
                   label = toupper(all_counties$NAME)),
               size = 2.5,
               color = 'grey40') +
  geom_point(aes(x = pb_long, y = pb_lat),
             color = pal[2],
             size = 2) +
  theme_minimal() +
  xlab('Longitude') +
  ylab('Latitude') +
  ggtitle('Lead')

#ggsave('~/capstone/co monitors.png')
```


### PM10
```{r}
pm10_lat <- unique(pm10$latitude)
pm10_long <- unique(pm10$longitude)
```

```{r}
pm10_monitors <- ggplot() +
  geom_sf(aes(geometry = all_counties$geometry),
          color = pal[10],
          fill = pal[10],
          alpha = .2,
          size = 1) +
  geom_sf_text(aes(geometry = all_counties$geometry,
                   label = toupper(all_counties$NAME)),
               size = 2.5,
               color = 'grey40') +
  geom_point(aes(x = pm10_long, y = pm10_lat),
             color = pal[2],
             size = 2) +
  theme_minimal() +
  xlab('Longitude') +
  ylab('Latitude') +
  ggtitle(expression(PM[10]))

#ggsave('~/capstone/co monitors.png')
```


### PM2.5
```{r}
pm2.5_lat <- unique(pm2.5$latitude)
pm2.5_long <- unique(pm2.5$longitude)
```

```{r}
pm2.5_monitors <- ggplot() +
  geom_sf(aes(geometry = all_counties$geometry),
          color = pal[10],
          fill = pal[10],
````

````
          alpha = .2,
          size = 1) +
  geom_sf_text(aes(geometry = all_counties$geometry,
                   label = toupper(all_counties$NAME)),
               size = 2.5,
               color = 'grey40') +
  geom_point(aes(x = pm2.5_long, y = pm2.5_lat),
             color = pal[2],
             size = 2) +
  theme_minimal() +
  xlab('Longitude') +
  ylab('Latitude') +
  ggtitle(expression(PM[2.5]))

#ggsave('~/capstone/co monitors.png')
```

### Sulfur Dioxide
```{r}
so2_lat <- unique(so2$latitude)
so2_long <- unique(so2$longitude)
```

```{r}
so2_monitors <- ggplot() +
  geom_sf(aes(geometry = all_counties$geometry),
          color = pal[10],
          fill = pal[10],
          alpha = .2,
          size = 1) +
  geom_sf_text(aes(geometry = all_counties$geometry,
                   label = toupper(all_counties$NAME)),
               size = 2.5,
               color = 'grey40') +
  geom_point(aes(x = so2_long, y = so2_lat),
             color = pal[2],
             size = 2) +
  theme_minimal() +
  xlab('Longitude') +
  ylab('Latitude') +
  ggtitle('Sulfur Dioxide')

#ggsave('~/capstone/co monitors.png')
```


### All maps together
```{r fig.height = 6}
monitor_locations <- grid.arrange(co_monitors, no2_monitors, ozone_monitors,
pb_monitors, pm10_monitors,
            pm2.5_monitors, so2_monitors, ncol = 2)
ggsave('~/capstone/monitor maps.png', monitor_locations)
```

### Monitor Table
```{r}
aqs_sites <- read.csv('~/capstone/air quality/aqs_monitors.csv') %>%
````

````
                          mutate(Last.Sample.Date =
as.Date(Last.Sample.Date))

```

```{r}
param_codes <- c(42101, 42602, 44201, 12128, 14129, 81102, 88101, 42401)
counties <- c('Allegheny', 'Armstrong', 'Beaver', 'Butler', 'Washington',
'Westmoreland')
aqs_filtered <- filter(aqs_sites, State.Code == '42' &
                             County.Name %in% counties &
                             Parameter.Code %in% param_codes &
                             Last.Sample.Date >= '2009-01-01')
```

```{r}
xtabs(~ County.Name + Parameter.Name, data = aqs_filtered)
```


```{r}
pb %>%
  group_by(county.name) %>%
  summarize(n = length(unique(site.num)))
```


## Full Models


```{r}
border1 <- fp_border(color="black", width = 1.5)
border2 <- fp_border(color="black", width = 1.25)
hyper_grid %>%
  arrange(min_logloss) %>%
  head(10) %>%
  rename("Learning Rate" = eta, "Minimum Node Size" = min_child_weight,
"Maximum Depth" = max_depth,
         "Optimal Trees" = optimal_trees, "Minimum Log Loss" = min_logloss)
%>%
  flextable() %>%
  font(fontname = 'Times New Roman', part = 'all') %>%
  hline_top(border = border1, part = 'header') %>%
  hline_bottom(border = border2, part = 'header') %>%
  hline_bottom(border = border1, part = 'body') %>%
  fontsize(size = 12, part = 'all')  %>%
  color(color = 'black', part = 'all') %>%
  autofit() %>%
  padding(padding = 1.5, part = 'all') %>%
  save_as_docx(path = "~/capstone/tuning table.docx")
```


```{r}
test_x <- select(test, -apgar5_cat.normal)
test_y <- test$apgar5_cat.normal
```
````
````

````
### XGB Full

```{r}
xgb_pred_prob_full <- predict(xgb_full, newdata = test_x, type = 'prob')
```


```{r}
cm_xgb_full <- confusionMatrix(data = xgb_pred_full, reference =
factor(test_y, levels = c(0,1), labels = c('abnormal', 'normal')))

```


```{r}
cm_xgb_full
```

#### Tuning

```{r fig.height = 4}
# png('~/capstone/tuning_results.png')
par(mfrow = c(8,1))
plot(xgb_full)
# dev.off()
```


```{r}
xgb_full$results %>%
  arrange(desc(Accuracy)) %>%
  select(-Kappa, -AccuracySD, -KappaSD) %>%
  head(1)
```






#### Variable Importance
```{r}
# create importance matrix
var_imp_full <- xgb.importance(model = xgb_full$finalModel)
```

```{r}
# variable importance plot
# labs <- c('Gestational Age', 'PM10 First', 'PM10 Third', 'SO2 Third',
'Ozone Second', 'NO2 Third',
#           'Lead Second', 'PM2.5 Third', 'PM2.5 First', 'Lead First',
"Mother's Age", 'Lead Third',
#           'NO2 Second', 'PM10 Second', 'CO First', 'Ozone First', 'SO2
Third', 'NO2 First', 'CO Third',
````
````

````
#           'CO Second')
gain_full <- xgb.ggplot.importance(var_imp_full, top_n = 20, measure =
"Gain") +
  theme_minimal() +
  ggtitle('Full Model') +
  scale_fill_manual(values = c(pal[4], pal[10]))

```


```{r}
cover_full <- xgb.ggplot.importance(var_imp_full, top_n = 20, measure =
"Cover") +
  theme_minimal() +
  ggtitle('Full Model') +
  scale_fill_manual(values = pal[10])
```







#### Partial Dependence
```{r}
library(pdp)
```



```{r}
partial_gest <- partial(xgb_full$finalModel, pred.var =
"gestational_age_weeks",
              plot = F,
              train = train_x,
              type = 'classification',
              prob = T)
``





```{r}
pdp1 <- ggplot(partial_gest, aes(x = gestational_age_weeks, y = yhat)) +
  geom_line(color = pal[10], size = .75) +
  theme_minimal() +
  xlab('Weeks') +
  ylab('Predicted Probability') +
  ylim(0,1) +
  ggtitle('Gestational Age')
```




```{r}
partial_pm10_first <- partial(xgb_full$finalModel, pred.var = "pm10_first",
              plot = F,
              train = train_x,
              type = 'classification',
````
````

````
              prob = T)
```

```{r}
pdp2 <- ggplot(partial_pm10_first, aes(x = pm10_first, y = yhat)) +
  geom_line(color = pal[10], size = .75) +
  theme_minimal() +
  xlab(expression("Concentration ("*mu*"g/m"^3*")")) +
  ylab('Predicted Probability') +
  ylim(0,1) +
  ggtitle(expression(PM[10]*' First Trimester'))
```


```{r}
partial_pm10_third <- partial(xgb_full$finalModel, pred.var = "pm10_third",
              plot = F,
              train = train_x,
              type = 'classification',
              prob = T)
```


```{r}
pdp3 <- ggplot(partial_pm10_third, aes(x = pm10_third, y = yhat)) +
  geom_line(color = pal[10], size = .75) +
  theme_minimal() +
  xlab(expression("Concentration ("*mu*"g/m"^3*")")) +
  ylab('Predicted Probability') +
  ylim(0,1) +
  ggtitle(expression(PM[10]*' Third Trimester'))
```




```{r}
partial_so2_second <- partial(xgb_full$finalModel, pred.var = "so2_second",
              plot = F,
              train = train_x,
              type = 'classification',
              prob = T,
              rug = T)
```

```{r}
pdp4 <- ggplot(partial_so2_second, aes(x = so2_second, y = yhat)) +
  geom_line(color = pal[10], size = .75) +
  theme_minimal() +
  xlab('Concentration (ppb)') +
  ylab('Predicted Probability') +
  ylim(0,1) +
  ggtitle(expression(SO[2]*' Second Trimester'))
```


```{r}
````

````
partial_ozone_second <- partial(xgb_full$finalModel, pred.var =
"ozone_second",
              plot = F,
              train = train_x,
              type = 'classification',
              prob = T)
```


```{r}
pdp5 <- ggplot(partial_ozone_second, aes(x = ozone_second, y = yhat)) +
  geom_line(color = pal[10], size = .75) +
  theme_minimal() +
  xlab("Concentration (ppm)") +
  ylab('Predicted Probability') +
  ylim(0,1) +
  ggtitle('Ozone Third Trimester')
```



```{r fig.height = 4}
pdp <- grid.arrange(pdp1, pdp2, pdp3, pdp4, pdp5)
ggsave('~/capstone/pdp.png', pdp)
```




### GLM Full
```{r}
border1 <- fp_border(color="black", width = 1.5)
border2 <- fp_border(color="black", width = 1.25)
as_flextable(glm_full$finalModel) %>%
  font(fontname = 'Times New Roman', part = 'all') %>%
  hline_top(border = border1, part = 'header') %>%
  hline_bottom(border = border2, part = 'header') %>%
  hline_bottom(border = border1, part = 'body') %>%
  fontsize(size = 12, part = 'all')  %>%
  color(color = 'black', part = 'all') %>%
  autofit() %>%
  padding(padding = 1.5, part = 'all') %>%
  save_as_docx(path = "~/capstone/glm_full_output.docx")
```



```{r}
cm_glm_full <- confusionMatrix(data = glm_pred_full, reference =
factor(test_y, levels = c(0,1), labels = c('abnormal', 'normal')))
cm_glm_full
```

### ROC

```{r}
glm_pred_prob_full <- predict(glm_full, newdata = test_x, type = 'prob')
````

````
xgb_pred_prob_full <- predict(xgb_full, newdata = test_x, type = 'prob')
```

```{r}
roc_glm_full <- roc(test_y, glm_pred_prob_full[,2])
roc_xgb_full <- roc(test_y, xgb_pred_prob_full[,1])
```

```{r}
ggroc(list(roc_glm_full, roc_xgb_full), size = .8) +
  scale_color_manual(labels = c(paste0('GLM: ', round(roc_glm_full$auc,3)),
                                paste0('XGB: ', round(roc_xgb_full$auc,3))),
                     values = c(pal[4], pal[10])) +
  labs(color = '') +
  theme_minimal()

ggsave('~/capstone/auc full.png')
```

## First Trimester Models

### GLM

```{r}
as_flextable(glm_first$finalModel) %>%
  font(fontname = 'Times New Roman', part = 'all') %>%
  hline_top(border = border1, part = 'header') %>%
  hline_bottom(border = border2, part = 'header') %>%
  hline_bottom(border = border1, part = 'body') %>%
  fontsize(size = 12, part = 'all')  %>%
  color(color = 'black', part = 'all') %>%
  autofit() %>%
  padding(padding = 1.5, part = 'all') %>%
  save_as_docx(path = "~/capstone/glm_first_output.docx")
```

```{r}
cm_glm_first <- confusionMatrix(data = glm_pred_first, reference =
factor(test_y, levels = c(0,1), labels = c('abnormal', 'normal')))
cm_glm_first
```

### XGB
```{r}
cm_xgb_first <- confusionMatrix(data = xgb_pred_first, reference =
factor(test_y, levels = c(0,1), labels = c('abnormal', 'normal')))
cm_xgb_first
````

````
```



#### Variable Importance
```{r}
var_imp_first <- xgb.importance(model = xgb_first$finalModel)
```

```{r}
gain_first <- xgb.ggplot.importance(var_imp_first, top_n = 20, measure =
"Gain") +
  theme_minimal() +
  ggtitle('First Trimester Model') +
  scale_fill_manual(values = c(pal[3],pal[4], pal[9], pal[10]))
```

```{r}
cover_first <- xgb.ggplot.importance(var_imp_first, top_n = 20, measure =
"Cover") +
  theme_minimal() +
  ggtitle('First Trimester Model') +
  scale_fill_manual(values = c(pal[4], pal[10]))
```






### ROC
```{r}
glm_pred_prob_first <- predict(glm_first, newdata = test_x, type = 'prob')
xgb_pred_prob_first <- predict(xgb_first, newdata = test_x, type = 'prob')
```


```{r}
roc_glm_first <- roc(test_y, glm_pred_prob_first[,1])
roc_xgb_first <- roc(test_y, xgb_pred_prob_first[,1])
```



```{r}
ggroc(list(roc_glm_first, roc_xgb_first), size = .8) +
  scale_color_manual(labels = c(paste0('GLM: ', round(roc_glm_first$auc,3)),
                                paste0('XGB: ', round(roc_xgb_first$auc,3))),
                     values = c(pal[4], pal[10])) +
  labs(color = '') +
  theme_minimal()

ggsave('~/capstone/auc first.png')
```
````
````

```
## Second Trimester Models

### GLM

```{r}
as_flextable(glm_second$finalModel) %>%
  font(fontname = 'Times New Roman', part = 'all') %>%
  hline_top(border = border1, part = 'header') %>%
  hline_bottom(border = border2, part = 'header') %>%
  hline_bottom(border = border1, part = 'body') %>%
  fontsize(size = 12, part = 'all')  %>%
  color(color = 'black', part = 'all') %>%
  autofit() %>%
  padding(padding = 1.5, part = 'all') %>%
  save_as_docx(path = "~/capstone/glm_second_output.docx")
```

```{r}
cm_glm_second <- confusionMatrix(data = glm_pred_second, reference =
factor(test_y, levels = c(0,1), labels = c('abnormal', 'normal')))
cm_glm_second
```


XGB
```{r}
cm_xgb_second <- confusionMatrix(data = xgb_pred_second, reference =
factor(test_y, levels = c(0,1), labels = c('abnormal', 'normal')))
cm_xgb_second
```


Variable Importance

```{r}
var_imp_second <- xgb.importance(model = xgb_second$finalModel)
```

```{r}
gain_second <- xgb.ggplot.importance(var_imp_second, top_n = 20, measure =
"Gain") +
  theme_minimal() +
  ggtitle('Second Trimester Model') +
  scale_fill_manual(values = c(pal[4], pal[10]))
```


```{r}
cover_second <- xgb.ggplot.importance(var_imp_second, top_n = 20, measure =
"Cover") +
  theme_minimal() +
  ggtitle('Second Trimester Model') +
  scale_fill_manual(values = c(pal[4], pal[9], pal[10]))
```
```

```
### ROC
```{r}
glm_pred_prob_second <- predict(glm_second, newdata = test_x, type = 'prob')
xgb_pred_prob_second <- predict(xgb_second, newdata = test_x, type = 'prob')
```


```{r}
roc_glm_second <- roc(test_y, glm_pred_prob_second[,1])
roc_xgb_second <- roc(test_y, xgb_pred_prob_second[,1])
```


```{r}
ggroc(list(roc_glm_second, roc_xgb_second), size = .8) +
  scale_color_manual(labels = c(paste0('GLM: ', round(roc_glm_second$auc,3)),
                                paste0('XGB: ',
round(roc_xgb_second$auc,3))),
                     values = c(pal[4], pal[10])) +
  labs(color = '') +
  theme_minimal()

ggsave('~/capstone/auc second.png')
```


```{r}
varImp_xgb_second <- varImp(xgb_second)[[1]] %>% slice_head(n=20)


varImp_xgb_second <- varImp_xgb_second %>%
  mutate(Var = factor(rownames(varImp_xgb_second))) %>%
  mutate(Var = fct_reorder(Var, Overall))

ggplot(varImp_xgb_second, aes(x = Overall, y = Var)) +
  geom_col(fill = pal[10]) +
  theme_minimal() +
  ggtitle('Feature Importance') +
  xlab('Importance') +
  ylab('')

ggsave('~/capstone/var imp second.png')
```


Third Trimester Models

GLM
```{r}
as_flextable(glm_third$finalModel) %>%
  font(fontname = 'Times New Roman', part = 'all') %>%
```
```

````
  hline_top(border = border1, part = 'header') %>%
  hline_bottom(border = border2, part = 'header') %>%
  hline_bottom(border = border1, part = 'body') %>%
  fontsize(size = 12, part = 'all')  %>%
  color(color = 'black', part = 'all') %>%
  autofit() %>%
  padding(padding = 1.5, part = 'all') %>%
  save_as_docx(path = "~/capstone/glm_third_output.docx")
```

```{r}
cm_glm_third <- confusionMatrix(data = glm_pred_third, reference =
factor(test_y, levels = c(0,1), labels = c('abnormal', 'normal')))
cm_glm_third
```

### XGB

```{r}
cm_xgb_third <- confusionMatrix(data = xgb_pred_third, reference =
factor(test_y, levels = c(0,1), labels = c('abnormal', 'normal')))
cm_xgb_third
```


#### Variable Importance
```{r}
var_imp_third <- xgb.importance(model = xgb_third$finalModel)
```

```{r}
gain_third <- xgb.ggplot.importance(var_imp_third, top_n = 20, measure =
"Gain") +
  theme_minimal() +
  ggtitle('Third Trimester Model') +
  scale_fill_manual(values = c(pal[3],pal[4], pal[9], pal[10]))
```

```{r}
cover_third <- xgb.ggplot.importance(var_imp_third, top_n = 20, measure =
"Cover") +
  theme_minimal() +
  ggtitle('Third Trimester Model') +
  scale_fill_manual(values = c(pal[4], pal[10]))
```




### ROC
```{r}
glm_pred_prob_third <- predict(glm_third, newdata = test_x, type = 'prob')
xgb_pred_prob_third <- predict(xgb_third, newdata = test_x, type = 'prob')
```


```{r}
````

````
roc_glm_third <- roc(test_y, glm_pred_prob_third[,1])
roc_xgb_third <- roc(test_y, xgb_pred_prob_third[,1])
```


```{r}
ggroc(list(roc_glm_third, roc_xgb_third), size = .8) +
  scale_color_manual(labels = c(paste0('GLM: ', round(roc_glm_third$auc,3)),
                                paste0('XGB: ', round(roc_xgb_third$auc,3))),
                     values = c(pal[4], pal[10])) +
  labs(color = '') +
  theme_minimal()

ggsave('~/capstone/auc third.png')
```



## All ROC together

```{r}
# ggroc(list(roc_xgb_full,
#             roc_glm_full,
#             roc_xgb_first,
#             roc_glm_first,
#             roc_xgb_second,
#             roc_glm_second,
#             roc_xgb_third,
#             roc_glm_third), size = .8) +
#   scale_color_manual(values = c(pal[1], pal[3:5], pal[7], pal[9:11])) +
#   labs(color = '') +
#   theme_minimal()

```


## All VarImp Together
```{r}
library(gridExtra)
```


### Gain

```{r fig.width=5}
varimp_gain <- grid.arrange(gain_full, gain_first, gain_second, gain_third)
ggsave('~/capstone/varimp_gain.png', varimp_gain)
```


### Cover
```{r fig.width=5}
varimp_cover <- grid.arrange(cover_full, cover_first, cover_second,
cover_third)
ggsave('~/capstone/varimp_cover.png', varimp_cover)
```
````
````

Bibliography

American Lung Association. (2021). American Lung Association. (2021). Most Polluted Places to Live. Retrieved from https://www.lung.org/research/sota/key-findings/most-polluted-places

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., . . . Yuan, J. (2022). xgboost: Extreme Gradient Boosting. Retrieved from https://CRAN.R-project.org/package=xgboost

Committee Opinion No. 644: The Apgar Score. (2015). *Obstetrics & Gynecology, 126*(4). Retrieved from https://journals.lww.com/greenjournal/Fulltext/2015/10000/Committee_Opinion_No__644__The_Apgar_Score.54.aspx

Ehrenstein, V. (2009). Association of Apgar scores with death and neurologic disability. *Clinical epidemiology, 1*, 45-53. doi:10.2147/clep.s4782

Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics, 29*(5), 1189-1232. Retrieved from http://www.jstor.org/stable/2699986

Health Effects Institute. (2020). *State of Global Air 2020*. Retrieved from Boston, MA: https://fundacionio.com/wp-content/uploads/2020/10/soga-2020-report.pdf

Lee, P.-C., Roberts, J. M., Catov, J. M., Talbott, E. O., & Ritz, B. (2013). First Trimester Exposure to Ambient Air Pollution, Pregnancy Complications and Adverse Birth Outcomes in Allegheny County, PA. *Maternal and Child Health Journal, 17*(3), 545-555. doi:10.1007/s10995-012-1028-5

Li, F., Wu, T., Lei, X., Zhang, H., Mao, M., & Zhang, J. (2013). The apgar score and infant mortality. *PloS one, 8*(7), e69072-e69072. doi:10.1371/journal.pone.0069072

Moster, D., Lie, R. T., Irgens, L. M., Bjerkedal, T., & Markestad, T. (2001). The association of Apgar score with subsequent death and cerebral palsy: A population-based study in term infants. *J Pediatr, 138*(6), 798-803. doi:10.1067/mpd.2001.114694

Naidoo, P., Naidoo, R. N., Ramkaran, P., & Chuturgoon, A. A. (2020). Effect of maternal HIV infection, BMI and NOx air pollution exposure on birth outcomes in South African pregnant women genotyped for the p53 Pro72Arg (rs1042522). *International Journal of Immunogenetics, 47*(5), 414-429. doi:https://doi.org/10.1111/iji.12481

Simon, L. V., Hashmi, M. F., & Bragg, B. N. (2021). *APGAR Score*: StatPearls Publishing, Treasure Island (FL).

US Environmental Protection Agency. (2019). Our Nation's Air. Retrieved from https://gispub.epa.gov/air/trendsreport/2019/documentation/AirTrends_Flyer.pdf

US Environmental Protection Agency. (2020). Our Nation's Air. Retrieved from https://gispub.epa.gov/air/trendsreport/2020/#home

US Environmental Protection Agency. (2021). Air Quality System Data Mart. Retrieved from https://www.epa.gov/outdoor-air-quality-data

Wei, H., Baktash, M. B., Zhang, R., wang, X., Zhang, M., Jiang, S., . . . Hu, W. (2021). Associations of maternal exposure to fine particulate matter constituents during pregnancy with Apgar score and duration of labor: A retrospective study in Guangzhou, China, 2012–2017. *Chemosphere, 273*, 128442. doi:https://doi.org/10.1016/j.chemosphere.2020.128442

World Health Organization. (2021). Ambient (outdoor) air pollution. Retrieved from https://www.who.int/health-topics/air-pollution#tab=tab_1