Prediction of Preterm Birth in Southwestern PA using Classification Models: A Comparative Analysis

by

Sabnum Pudasainy

BS, The University of Louisiana at Monroe, 2020

Submitted to the Graduate Faculty of the

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

Sabnum Pudasainy

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 Preterm Birth in Southwestern PA using Classification Models: A Comparative Analysis

Sabnum Pudasainy, MS

University of Pittsburgh, 2022

Background: Preterm birth is a global health burden and a leading cause of neonatal mortality and morbidity. This study aims to compare prediction models to identify clinical, demographic, and environmental risk factors associated with preterm birth using binary classification methods.

Methods: Data from 221,060 infants born between 2010 and 2020 to mothers who resided in eight southwestern Pennsylvania counties (Allegheny, Armstrong, Beaver, Butler, Fayette, Greene, Washington, Westmoreland) were used. Covariates utilized for this analysis were the mother's and the neonate's clinical and demographic features and the mother's mean exposure to air pollutants - Carbon monoxide (CO), Nitrogen dioxide (NO_2), Particulate Matter ($PM_{2.5}$), Ozone (O_3) and Sulfur dioxide (SO_2) in mother's geocoded areas of residence during the mother's gestation period. Exploratory data analysis, including Empirical Bayes approach, was conducted to better understand the covariates and the outcome, i.e., preterm birth. Further, three supervised machine learning techniques – Elastic Net (GLMNET), Support Vector Machine (SVM) and Random Forest – were used to build and compare prediction models based on performance metrices like Area under the Curve (AUC), sensitivity and specificity.

Results: Empirical Bayes identified mothers with fewer prenatal visits (0-10) and mothers who resided in Allegheny County to be associated with higher posterior average for event probability. Among the three different algorithms used to predict preterm birth, Random Forest seemed to outperform GLMNET and SVM with an AUC of 0.83, compared to 0.77 for both

GLMNET and SVM. The top important predictors common to GLMNET and SVM were total number of prenatal visits, mother's race and education. Additionally, Random Forest identified mean exposures to pollutants as the top features, along with number of prenatal visits and Allegheny as the mother's residential county. The results from Empirical Bayes exploration and the classification models were fairly consistent.

Public Health Significance: Optimal prediction of preterm birth facilitates early identification and treatment of at-risk mothers, and enables targeted interventions to minimize infant mortality and morbidity, which would significantly benefit the community, nation, and the healthcare system as a whole. The environmental factors identified here should be explored further.

Table of Contents

List of Tables

List of Figures

1.0 Introduction

1.1 Background and Significance

Preterm birth, as defined by the World Health Organization, is any birth that takes place before 37 weeks of gestation (World Health Organization, n.d.). According to an estimated distribution of causes of 3.1 million neonatal deaths in 193 countries in 2010, preterm birth directly contributed to about 35% of all neonatal deaths, and indirectly contributed to an increased chance of post neonatal deaths, especially deaths from neonatal infections (Blencowe et al., 2013). In the United States, preterm birth represented 10.1% of live births in 2020, i.e., 1 in 10 infants were born preterm. Over 26 billion dollars is spent annually for the preterm deliveries in the United States (March of Dimes, n.d.). Ability to predict preterm births accurately would enable clinicians to identify and treat at-risk mothers on time. It would allow clinicians to utilize targeted interventions to reduce the burden of preterm birth.

1.2 Risk factors for preterm birth

To build a preterm birth risk prediction model, a better understanding of risk factors that affect preterm birth is crucial. Previous studies have shown relationship of preterm birth with maternal demographic characteristics like mother's race, education, age, socio-economic status, biologic and genetic markers, nutritional status, pregnancy history, smoking status, intrauterine infection etc. For instance, women from the black population had higher rates of preterm births

(16-18%) compared to white women (5-9%) (Goldberg et al., 2008). Other demographic factors like mother's lower educational attainment and lower socioeconomic status (Goldberg et al., 2008), mothers younger than 18 and older than 35 (Martin et al., 2018) were also found to be associated with preterm birth. Use of tobacco was found to increase the risk of preterm births (< 2-fold) after adjusting for other factors (Goldberg et al., 2008).

A systematic review of 68 studies that looked at over 32 million births in the United States reported that exposure to fine particulate matter ($PM_{2.5}$) and ozone (O_3) was linked to higher risk for preterm births, i.e., the risk of preterm birth increased by a median of 11.5% when exposed to $PM_{2.5}$ and the risk increased from 3% to 9.6% when exposed to O_3 (Bekkar et al., 2020). A time-series analysis done in Pennsylvania from 1997-2001 found increased risk of preterm birth with exposure to PM_{10} (RR=1.07 per 50 $\mu g/m^3$ increase) and SO_2 (RR = 1.15 per 15 ppb increase) in the six weeks before birth (Sagiv et al., 2005). This analysis aims to investigate if mother's exposure to air pollutants during the gestation period, in addition to maternal demographic factors, is associated with preterm birth. The air pollutants being studied are Carbon monoxide (CO), Nitrogen dioxide (NO_2), Ozone (O_3), particulate matter less than 2.5 microns ($PM_{2.5}$) and Sulfur dioxide (SO_2).

1.3 Objectives

With current advents in technology for collection and storage of complex and high-volume medical data, there are many opportunities for us to explore novel relationships between health risks and outcome. Increased interest has been seen over the last few years regarding the use of Artificial Intelligence (AI) techniques in medical sector and especially in reproductive health. One

subcategory of AI is Machine Learning (ML), which uses complex algorithms to find any potential links in the data and provides insights for making clinical decisions (Wang et al., 2019).

The primary aim of this analysis is to build three classification models using supervised ML algorithms to identify potential risk factors associated with preterm birth, including exposure to air pollutants, and compare the predictive performance of these models. The other aim is to evaluate the importance of predictors used to build the best performing model and to identify the predictors which are most significant in optimal prediction of preterm birth.

2.0 Methods

The framework that was followed for data analysis is shown in Figure 1.

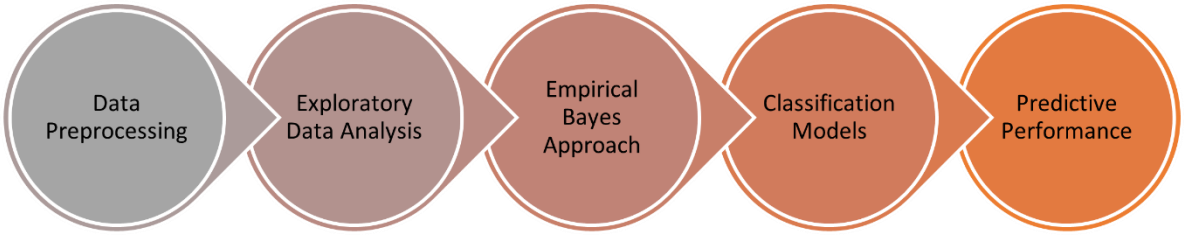

Figure 1 Data Analysis Framework

2.1 Data

2.1.1 Birth Data

Birth data were retrieved from the Bureau of Health Statistics and Research, Department of Health, Pennsylvania for years 2010 to 2020. Inclusion criteria for the birth records data were as follows:

i. Child Date of Birth between Jan 1, 2010, and Dec 31, 2020
ii. Maternal residence within the eight Southwest PA counties (Allegheny, Armstrong, Beaver, Butler, Fayette, Greene, Washington, Westmoreland)

Exclusion criteria for the birth records data were as follows:

i. Multiple births (non-singleton births)

ii. Stillbirth

iii. Birth weight < 500g

iv. Gestational age < 22 weeks

All clinical and demographic features of the neonate and mother were treated as categorical.

Neonate sex had two levels: male and female.

Neonate's date of birth was used to calculate the season of birth for the neonate: Spring (March, April, May), Summer (June, July, August), Autumn (September, October, November) and Winter (December, January, February).

Mother's residential county had eight levels: Allegheny, Armstrong, Beaver, Butler, Fayette, Greene, Washington, and Westmoreland. The counties with less than 5% frequency (Fayette, Armstrong, and Greene) were combined as "Other".

Mother's education had eight levels: 8th grade or less, 9th-12th grade/No diploma, High school graduate/GED completed, Some college credit but not a degree, Associate degree, Bachelor's degree, Master's degree, and Doctorate/Professional degree. Mother's education categories were collapsed into five levels: Less than HS, HS/GED/Some college, Associate degree, Bachelor's degree, and Graduate/Professional degree.

Mother's age was a continuous variable. It was categorized into six levels: <20, 20-24, 25-29, 30-34, 35-39 and 40+.

Mother's race had 15 levels: White, Black/African American, American Indian/Alaska Native, Asian Indian, Chinese, Filipino, Japanese, Korean, Vietnamese, Other Asian, Native Hawaiian, Guamanian/Chamorro, Samoan, Other Pacific Islander, Other. Mother's race categories were collapsed into five levels: White, Black/African American, American Indian/Alaska Native,

Asian/PI and Other. However, due to frequency less than 5%, these categories were further lumped into 3 categories: White, Black/African American and Other.

Mothers' height and pre-pregnancy weight were used to calculate pre-pregnancy Body Mass index (BMI). It was then categorized into four levels: Underweight (<18.5), Normal (18.5-24.9), Overweight (25-29.9) and Obese (30 and above).

Mother's receipt of WIC services, which is a surrogate for low family socioeconomic status, had two levels: yes and no.

Mother's diagnosis of gestational diabetes, which is a risk factor in this pregnancy, had two levels: yes and no.

Mother's smoking status prior to pregnancy and during the three trimesters were dichotomized into yes and no. Smoking status during the three trimesters were combined to create a new smoking variable. The combinations with less than 5% frequency were lumped resulting in a total of 3 categories: Yes-Yes-Yes, No-No-No and Other. However, to use both smoking prior to pregnancy and the combined smoking variable during the three trimesters was redundant, as they followed similar distribution. Hence, only smoking prior to pregnancy was used for analysis.

Mother's previous live births had 14 levels: 0-14. Because of the low frequency of some of the levels, lumping was used to create five levels: 0, 1, 2, 3 and Other.

The total number of prenatal visits for mother was a continuous variable. It was used to represent mother's access to prenatal care and was categorized into three levels: 0-10, 10-20 and 20+.

For each child, the period of gestation was calculated using the child's date of birth and obstetric estimate of gestation (in weeks).

2.1.2 Exposure Data

The environmental exposure data were obtained from the Air Quality System (AQS) data provided by the United States Environmental Protection Agency (US EPA).

The data for each of the pollutants - Carbon monoxide (CO), Nitrogen dioxide (NO_2), Particulate Matter ($PM_{2.5}$), Ozone (O_3) and Sulfur dioxide (SO_2) - from the year 2009-2020 was downloaded from the EPA website and were filtered to include the eight southwestern PA counties. O_3 and CO were measured in parts per million (ppm), SO_2 and NO_2 were measured in parts per billion (ppb) and $PM_{2.5}$ was measured in micrograms per cubic meter ($\mu g/m^3$). To calculate mother's mean exposure to pollutants, all the active monitor sites during the mother's gestation period were identified. Then, the distances between mother's residence and the active monitors were calculated using geocoded latitudes and longitudes to identify the nearest monitor. Further, all the daily average concentrations during the mother's gestation period for the particular pollutant were extracted based on the nearest monitor's records and the mean was calculated. This mean concentration was assigned as mother's mean exposure to pollutants during the gestation period. The mean and median exposures were highly correlated, so only the mean exposure was used for analysis.

The maps with active EPA monitor sites within each of the eight southwest Allegheny counties for CO, NO_2, O_3, $PM2_{.5}$ and SO_2 are shown in Figure 2. The maps were made using R's *ggmap* library.

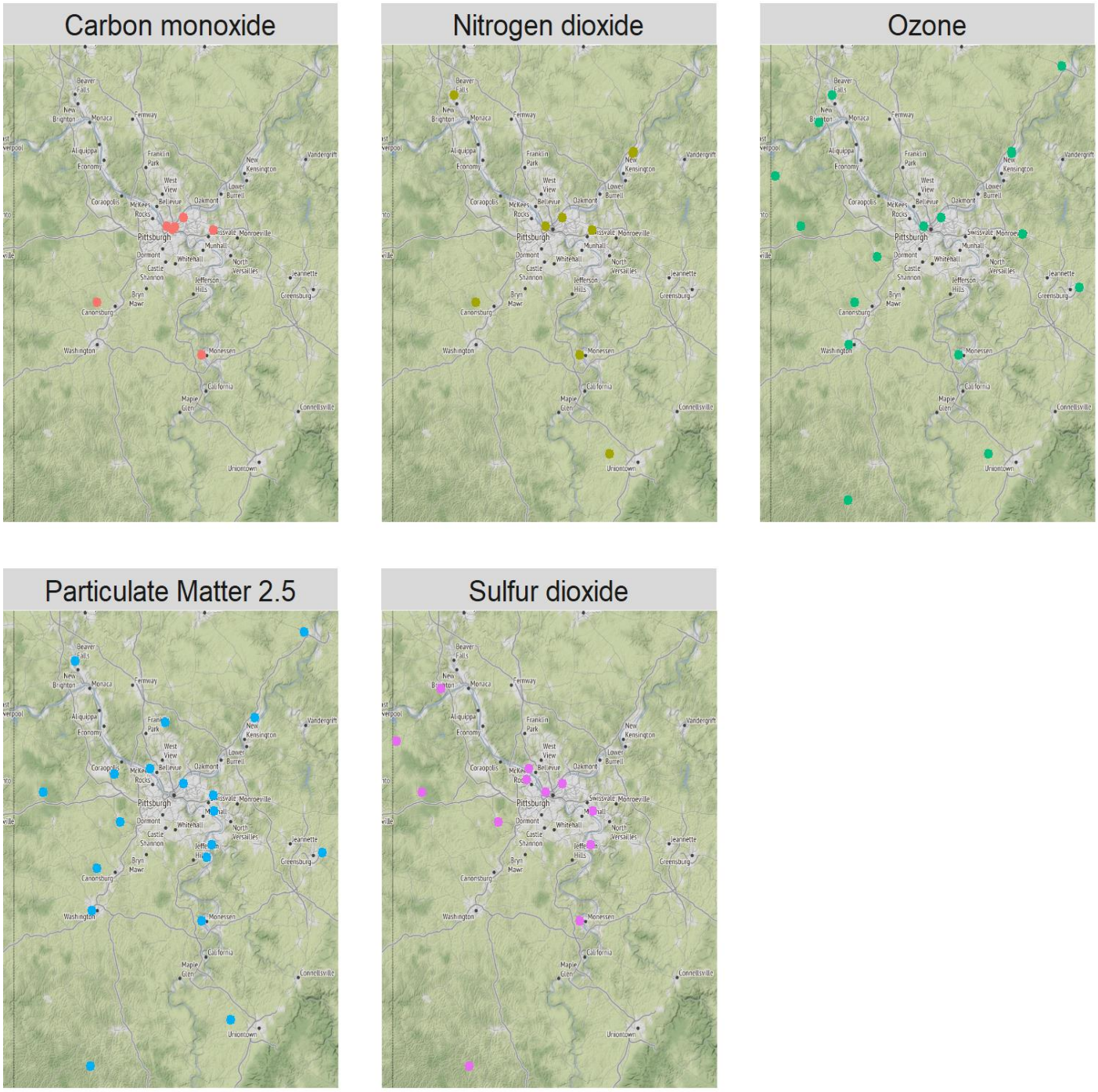

Figure 2 Map plots for EPA monitor sites

2.1.3 Outcome and covariates

Outcome: Based on obstetric estimate of gestation, preterm births was classified as "yes" for births occurring at < 37 weeks of gestation and "no" for births occurring at >= 37 weeks of gestation.

Covariates: Variables that were utilized for analysis were as follows:

i. Clinical and demographic features of the neonate and mother: Neonate sex, season of neonate birth, mother's residential county, mother's education, mother's age, mother's race, mother's receipt of WIC services, mother's diagnosis of gestational diabetes, smoking prior to pregnancy, number of previous live births for mother, number of prenatal visits.

ii. Environmental exposures: Mother's mean exposure to air pollutants (CO, NO_2, O_3, $PM2_{.5}$ and SO_2) in mother's geocoded areas of residence during the mother's gestation period.

2.1.4 Data preprocessing

The data was then assessed for missing value. There were 1,932 records with missing values for preterm births. These records were excluded from the study. The data was filtered to include only those births where mother's residential latitude and longitude were known. 8,153 records with missing values for latitude and longitude were dropped. Missing data visualization for the data after preprocessing is shown in Figure 3.

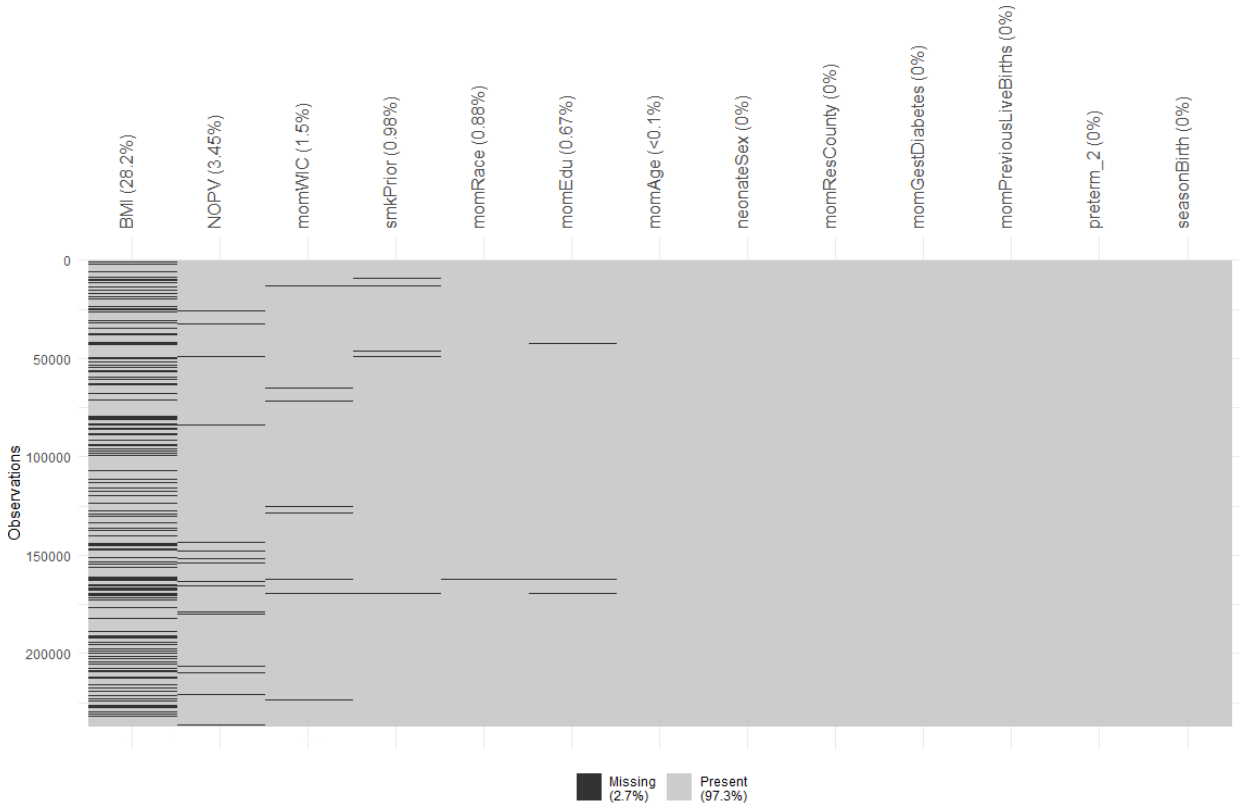

Figure 3 Missing Data Visualization

As seen in Figure 3, BMI had about 28% of missing data, because of which BMI was dropped from the set of predictor variables. The missingness for other variables did not seem concerning and were be assumed to be missing completely at random. These observations with missing values were dropped from the analysis.

2.2 Analysis section

2.2.1 Exploratory Data Analysis

The categorical and continuous predictor variables by preterm births were explored using bar plots and frequency polygons respectively. To further understand the categorical variables, an Empirical Bayes exploration was carried out.

2.2.1.1 Empirical Bayes Approach

Before predictive modeling, Empirical Bayes was used to explore the categorical variables of the data. The unique possible combinations of all the categories were used to represent the groups. Empirical Bayes can only be used when there are a large number of groups and since there were 40,715 groups, the use of Empirical Bayes was valid.

For each group, the number of preterm births and total births were summarized, and proportion of preterm births was calculated. This proportion is the Maximum Likelihood Estimate (MLE) for the event probability. There were many groups for which the total number of births were very low. For instance, there were 20,280 groups which had one total birth. To get a better estimate, these noisy data were removed, and only groups with at least 50 births were selected to craft the informative prior. The cutoff point was selected such that it prevented small sample sizes from yielding unreliable estimates, but still allowed a variety in the value of MLEs. There were 568 groups with at least 50 births in the data. The estimated beta prior is shown by the red line in Figure 4.

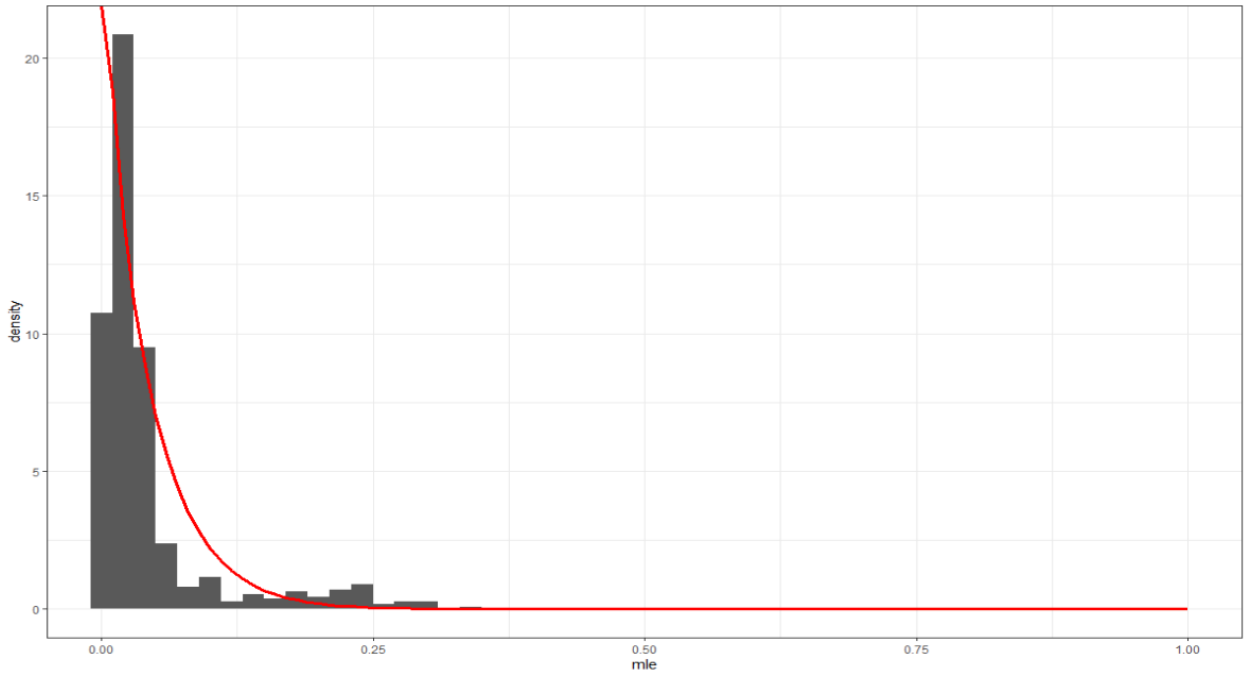

Figure 4 Informative prior for all groups with at least 50 births

It can be seen from Figure 4 that the prior is preventing high event probabilities. The other finding is that there appears to be two modes, one near 0 and the other near 0.25. The distribution of the event probability for all groups with at least 50 births could be a mixture of betas. However, just for exploration purposes, the empirical Bayes beta prior was applied to the beta-binomial likelihood for all groups. Each group had a beta distribution as the posterior. The distribution of the posterior means across groups is shown in Figure 5. Three modes are depicted in Figure 5, first one is less than 5%, the second one is near 7%, and the third one is a little above 10%. The posterior mean distribution in Figure 5 displays that a small number of groups have rather high posterior means with values above 15%.

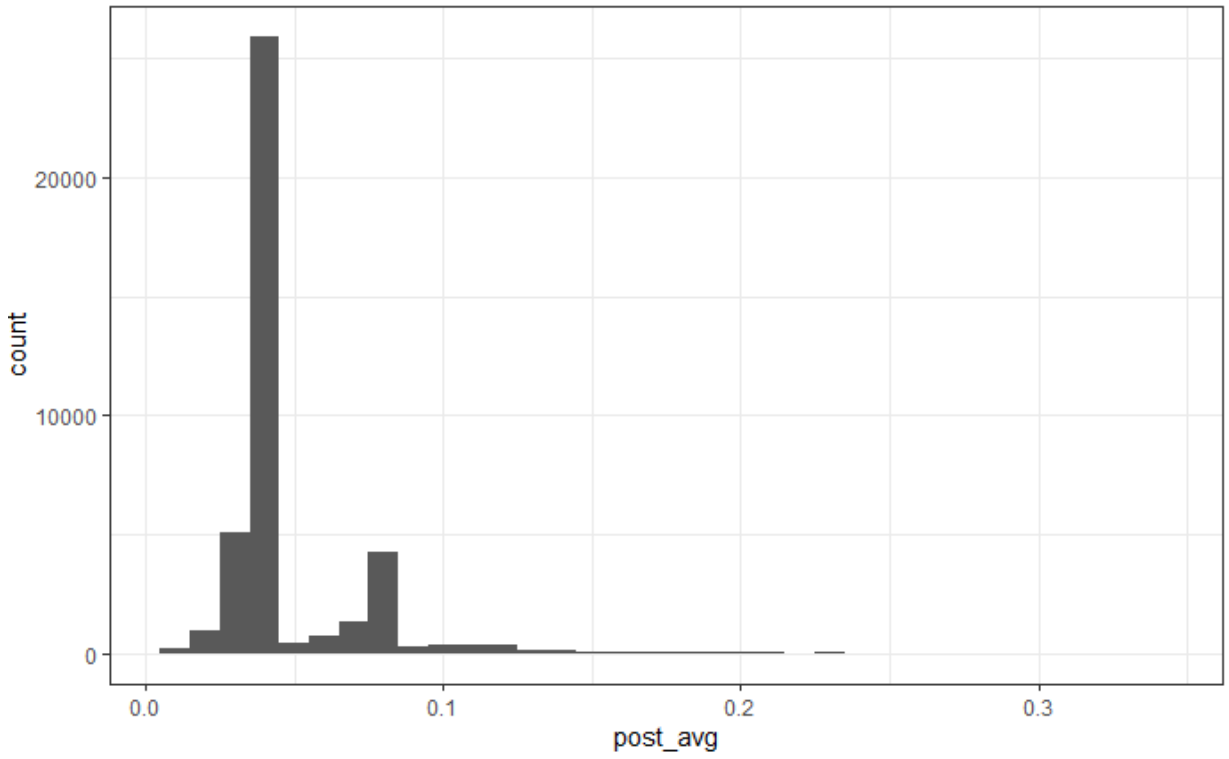

Figure 5 Distribution of posterior means for all groups with at least 50 births

Finally, the distribution of the posterior means conditioned on the individual categorical variables was plotted to see which variable was associated with higher posterior means.

2.2.2 Train Test Split

Data splitting helps us understand how well the model generalizes to unseen or future data. A training set is used in the model building process, in which the features are used to train a model that accurately predicts the outcome. After a model is selected from the training set, a testing set is used to evaluate unbiased model performance.

After data preprocessing and exploration, the dataset was split into training and testing sets. Some of the commonly used train-test split include splitting the data in the ratio 60-40, 70-30 or

80-20. While choosing the split, if too much data are used to train (e.g., more than 80%), the model would fit the training data very well but would not generalize well to the other data. This leads to overfitting. On the other hand, if too much data are used to test (e.g., more than 40%), we would not get a good assessment of model parameters (Boehmke et al., 2020). Additionally, if too much data are used to test at the expense of training set, we are evaluating a model that did not reach its full learning capacity because of the lack of data in the training dataset. Therefore, an optimal selection of train-test split is vital to learning from the data. For this analysis, a train-test split in the ratio of 70-30 was used.

2.2.2.1 Resampling and sub-sampling methods

The outcome variable in the dataset, preterm births, was heavily imbalanced. In both the training and testing sets, the proportion of term birth and preterm birth were about 0.927 and 0.07 respectively. Imbalanced dataset poses great challenge on model performance because it introduces a prediction bias for the abundant class (Leevy et al., 2018). To handle imbalanced data, down-sampling technique was utilized. Down-sampling reduces the size of the more frequent class to match the occurrence of the less frequent class. As a result, the two outcome classes are balanced on the dataset. Down-sampling was performed on the training set before building the models using R's *ROSE* package.

Down-sampling was used in this analysis for two main reasons:

i. The dataset was huge containing 11,004 preterm births i.e., there was sufficient data for the analysis (Boehmke et al., 2020).
ii. Down-sampling greatly reduced the computation time compared to other sub-sampling methods like Synthetic Minority Oversampling Technique.

The resampling method utilized while training the models was *k*-fold cross validation. *k*-fold cross-validation divides the training set into *k* folds of roughly same size, such that the model is trained on *k*-1 folds and evaluated on the last remaining fold. The process is repeated *k* times and each time, a different fold is used to assess the model performance. This results in *k* performance values, which are then averaged to compute the overall model behavior (Boehmke et al., 2020).

In this analysis, a repeated cross validation with 5-folds and 3 repeats was utilized using R's *caret* package. This method performs a 5-fold cross-validation on the training data 3 times, and for each cross-validation, a different set of folds is utilized. Because of this, a repeated cross-validation helps to improve the estimates of the model performance.

2.2.3 Classification model: Elastic Net

Elastic Net is a regularization technique that combines the penalties from the ridge and lasso (least absolute shrinkage and selection operator) regression. The goal for a regularized regression model is to minimize the Sum of Squared Errors (similar to Ordinary Least Squares), in addition to a penalty term P.

$$minimize(SSE + P) \quad \text{(Eq 1)}$$

The penalty parameter constrains the size of coefficients, and the coefficients can increase only when there is a comparable decrease in the value of SSE (Boehmke et al., 2020).

Ridge regression pushes the correlated features towards one another and shrinks the coefficient estimates for the less important features to approximately zero. However, it does not perform feature selection, meaning it will still retain all the original features in the model. Lasso

overcomes this limitation by shrinking the coefficient estimates for the less important features to exactly zero, thus conducting automated feature selection.

Elastic Net takes the best of both worlds such that it provides with ridge penalty's effective regularization (especially for correlated features) and lasso penalty's feature selection characteristics.

$$minimize(SSE + \lambda_1 \sum_{j=1}^{p} \beta_j^2 + \lambda_2 \sum_{j=1}^{p} |\beta_j|) \quad \text{(Eq 2)}$$

β_j represents coefficient estimate for the jth feature and p represents the total number of features. The first penalty term in the above equation comes from the ridge regression, known as L2-regularization, which penalizes the sum of squares of the estimates. The second penalty term comes from lasso, known as L1-regularization, which penalizes the sum of absolute values of the estimates. Lambda (λ) is a shrinkage parameter that controls the penalties. For instance, when lambda is sufficiently large, coefficients are shrunk strongly which forces some coefficients to be exactly zero in lasso (James et al., 2021).

R *caret's glmnet* method was used to perform this analysis. Alpha (α) is an argument in *glmnet* method which when set to 0 indicates a ridge regression model and when set to 1 indicates a lasso model. A tuning grid with alpha 0.1, 0.2, 0.3, 04 and lambda as a sequence of exponent of 21 numbers from -6 to 1 was used to tune the parameters - alpha and lambda. The model identified best parameters as alpha of 0.2 and lambda of 0.003517517.

2.2.4 Classification model: Support Vector Machine (SVM)

Support Vector Machine in binary classification is based on the fundamental idea of finding a hyperplane that best divides the data into the two classes/categories. Terminologies to better understand the SVM algorithm are listed as follows:

i. Support Vectors: The data points closest to the hyperplane in both the classes

ii. Margin: The distance between the hyperplane and the support vectors

iii. Hyperplane: A hyperplane is defined as

$$f(X) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p = 0 \quad \text{(Eq 3)}$$

This linear equation defines hyperplane in p-dimensional feature space. β_0 represents the intercept, β_1 represents the coefficient for X_1 and β_p represents the coefficient for X_p. When p=2, the hyperplane is a line in 2-D space. When p=3, the hyperplane is a plane in 3-D space.

The optimal hyperplane is the one for which the margin is maximized. However, with real-world data, it might not always be feasible to find a hyperplane that accurately separates the classes using the original features. SVM overcomes this obstacle by mapping the original feature space or non-linear data into a higher dimension space so that data can be linearly divided by a plane. This is called the kernel trick. In the enlarged or kernel-induced feature space, SVM then finds a hyperplane to separate the classes. The decision boundary from the enlarged feature space is then projected back to the original feature space.

SVM uses kernel functions like d-th degree polynomial, radial basis function, hyperbolic tangent to enlarge the feature space. For this analysis, radial basis function was used because of its high flexibility.

$$K(x, x') = \exp(\gamma||x - x'||^2) \quad \text{(Eq 4)}$$

The parameters for a SVM Radial Basis Function (RBF) kernel are C and gamma. The gamma parameter is related to the inverse of the sigma parameter of a normal distribution (Boehmke et al., 2020). Gamma defines the curvature in the decision boundary. For instance, when

gamma is low, the decision surface is very broad. The parameter C penalizes misclassification of a data point against the simplicity of decision surface. A lower C value means that the model is fine with misclassified data points, whereas a higher C value means the model aims to classify all data points correctly (*Support Vector Machines*, n.d.).

SVM was implemented using R's *caret* package. The default parameters selected by the model were sigma of 0.01330383 and C of 0.25.

2.2.5 Classification model: Random Forest (RF)

Random forest builds on the principles of decision tress and bagging. Bagging aggregates the predictions across all the decision trees built on bootstrapped copies of the training data, but random forest takes it a step further and performs split-variable randomization. This means, for every split or node of the decision tree, a random subset of mtry of original p features is utilized. The typical default value used for mtry in classification is $\sqrt{p}$. This introduces more randomness into the tree building process and helps to reduce tree correlation, which is a limitation of the bagging method (Boehmke et al., 2020).

Random forest uses Gini index for splitting the nodes of the decision tree. For any classification problem, Gini index of a node (n) is given as

$$Gini(n) = 1 - \sum_{j=1}^{c}(p_j)^2 \quad \text{(Eq 5)}$$

where c is the number of classes (c=2 for a binary classification) and p_j is the relative frequency of the class being observed for that node n. The higher the decrease in Gini score, the higher is the importance of the variable in dividing the data into two classes (Sarica et al., 2017).

The increasing popularity of Random Forests is due to its decent out-of-the-box performance (Boehmke et al., 2020), which is why only the default tuning parameters were used for this analysis. Random Forest was implemented using R's *caret* package.

2.2.6 Performance metrics

Accuracy as a performance metric is not always adequate. For instance, when the number of non-events is much higher than the number of events (Kubat et al., 1998). The metrics that were used to compare the performance of the models in both training and testing sets were sensitivity and specificity and AUC (Area Under the Curve). In addition, accuracy was also compared for the model performance on testing sets.

For a 2-class classification model, performance metrics are based on the confusion matrix, which is shown as follows:

Confusion Matrix	**Predicted Class**	
True Class	**Event**	**Non-event**
Event	True Positive (TP)	False Negative (FN)
Non-event	False Positive (FP)	True Negative (TN)

Sensitivity (True Positive Rate) is the proportion of events (preterm births) that are correctly identified.

Sensitivity = TP / (TP+FN)

Specificity (True Negative Rate) is the proportion of non-events (term births) that are correctly identified.

Specificity = TN / (FP+TN)

AUC is the area under the Receiver Operating Characteristic (ROC) curve which plots sensitivity (True Positive Rate) along the Y-axis and 1-specificity (False Positive Rate) along the X-axis. It represents the diagnostic ability of the classifier to distinguish between the two outcome classes. AUC value lies between 0.5 to 1, and higher the AUC, the better the performance of the model in distinguishing these two classes.

Accuracy is the proportion of correct predictions made by the classifier.

Accuracy = (TP+TN) / (TP+FN+FP+TN)

The metric “ROC” was used in the caret package to assess performance of the models in both training and testing datasets.

3.0 Results

Analysis results are presented in two sub-sections. Section 3.1 includes a brief characterization of the study cohort with visualizations for distribution of categorical and continuous variables by preterm birth. Section 3.2 contains subsections with results from the Empirical Bayes exploration and supervised machine learning models on the training and testing datasets.

3.1 Descriptive Statistics

Descriptive outcome and covariate statistics for the study population of 221,060 births are included Table 1 (for categorical covariates) and Table 2 (for continuous covariates).

The distribution of preterm birth and term birth looks similar for majority of the categorical covariates, i.e., ~93% term births and ~7% preterm births in the dataset. However, the proportion of preterm births is comparatively higher for some of the levels within the categorical covariates. For instance, mothers with more than three previous live births, mothers with less than HS education, mothers who are 40 years or older, Black or African American mothers, mothers who had gestational diabetes and mothers who smoked prior to pregnancy have higher proportion of preterm births than the dataset. The most striking difference though, is for mothers who had 0-10 prenatal visits. This level had 18.4% preterm births which is much higher than the other categories and their levels. This information is visually depicted in Figure 6. The distribution of preterm birth

and term birth for each of the mean pollutant exposure seem to be similar for most parts. This can be seen in Figure 7.

Table 1 Descriptive outcome and categorical covariate statistics

Categorical covariates	Levels	Preterm births N (%)	Term births N (%)
Number of previous live births			
	00	7017 (7.4%)	87958 (92.6%)
	01	4591 (6.1%)	70178 (93.9%)
	02	2422 (7.4%)	30179 (92.6%)
	03	1038 (8.9%)	10584 (91.1%)
	Other	807 (11.4%)	6286 (88.6%)
Mother's education			
	Associate degree	1782 (7.6%)	21808 (92.4%)
	Bachelor's degree	3315 (5.7%)	55195 (94.3%)
	Graduate/Professional Degree	2309 (5.6%)	38843 (94.4%)
	HS/GED/Some college	6931 (8.4%)	75511 (91.6%)
	Less than HS	1538 (10.0%)	13828 (90.0%)
Mother's age			
	<20	864 (8.8%)	8943 (91.2%)
	20-24	2951 (7.9%)	34618 (92.1%)
	25-29	4371 (6.7%)	60481 (93.3%)
	30-34	4685 (6.5%)	67167 (93.5%)
	35-39	2421 (7.8%)	28737 (92.2%)
	40+	583 (10.0%)	5239 (90.0%)
Mother's race			
	Black or African American	3130 (10.7%)	25992 (89.3%)
	White	11936 (6.6%)	167931 (93.4%)
	Other	809 (6.7%)	11262 (93.3%)
Mother's residential county			
	Allegheny	8934 (7.3%)	113298 (92.7%)
	Beaver	1001 (6.4%)	14596 (93.6%)
	Butler	1024 (6.2%)	15596 (93.8%)
	Washington	1351 (7.1%)	17582 (92.9%)
	Westmoreland	1992 (7.0%)	26276 (93.0%)
	Other	1573 (8.1%)	17837 (91.9%)
Receipt of WIC services			

Categorical covariates	Levels	Preterm births N (%)	Term births N (%)
	No	10789 (6.8%)	147921 (93.2%)
	Yes	5086 (8.2%)	57264 (91.8%)
Number of prenatal visits			
	0-10	11208 (18.4%)	49800 (81.6%)
	11-20	4354 (2.8%)	150767 (97.2%)
	20+	313 (6.3%)	4618 (93.7%)
Gestational diabetes			
	No	14872 (7.1%)	195551 (92.9%)
	Yes	1003 (9.4%)	9634 (90.6%)
Season of birth			
	Autumn	3842 (7.0%)	51286 (93.0%)
	Spring	4069 (7.3%)	51480 (92.7%)
	Summer	4234 (7.2%)	54839 (92.8%)
	Winter	3730 (7.3%)	47580 (92.7%)
Neonate sex			
	Female	7177 (6.6%)	100815 (93.4%)
	Male	8698 (7.7%)	104370 (92.3%)
Smoking prior to pregnancy			
	No	12090 (6.7%)	169507 (93.3%)
	Yes	3785 (9.6%)	35678 (90.4%)

Table 2 Descriptive outcome and continuous covariate statistics

Mean Exposure to pollutants (ppm)	Preterm births Mean (SD)	Term births Mean (SD)
Carbon Monoxide	0.32 (0.12)	0.31 (0.11)
Nitrogen Dioxide	9.43 (3.18)	9.36 (3.22)
Ozone	0.03 (0.00)	0.03 (0.00)
Particulate Matter 2.5	10.23 (1.95)	10.13 (1.91)
Sulfur Dioxide	2.24 (1.89)	2.21 (1.88)

Distribution of categorical variables by preterm births is also shown in Figure 6 as stacked bar plots.

Figure 6 Distribution of categorical variables by preterm births

Distribution of continuous variables by preterm births is shown in Figure 7 as histograms/frequency polygons.

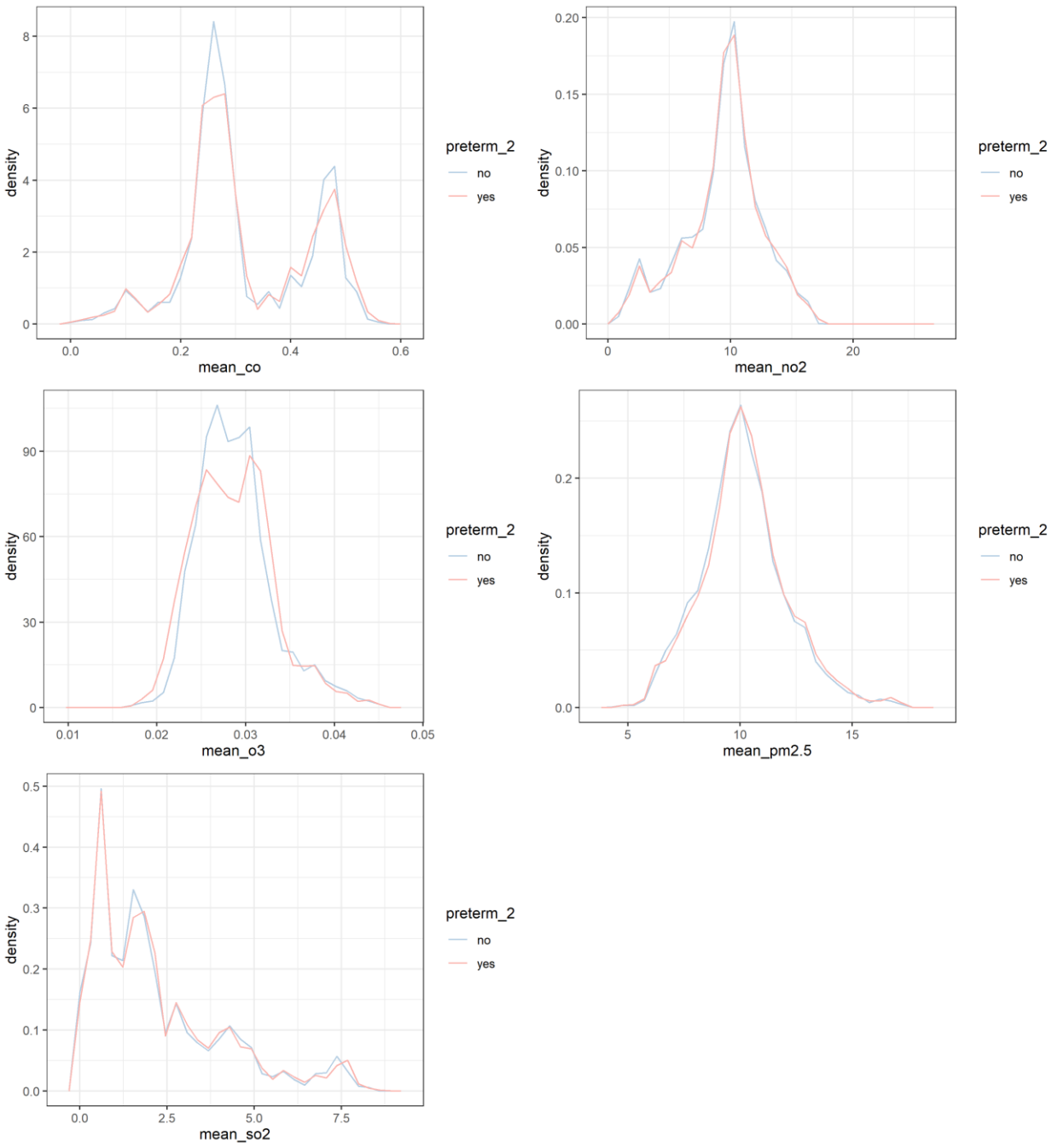

Figure 7 Distribution of continuous variables by preterm births

3.2 Machine Learning Results

3.2.1 Empirical Bayes to explore the categorical variables

The distribution of the posterior means conditioned on the individual categorical variables is depicted in Figure 8.

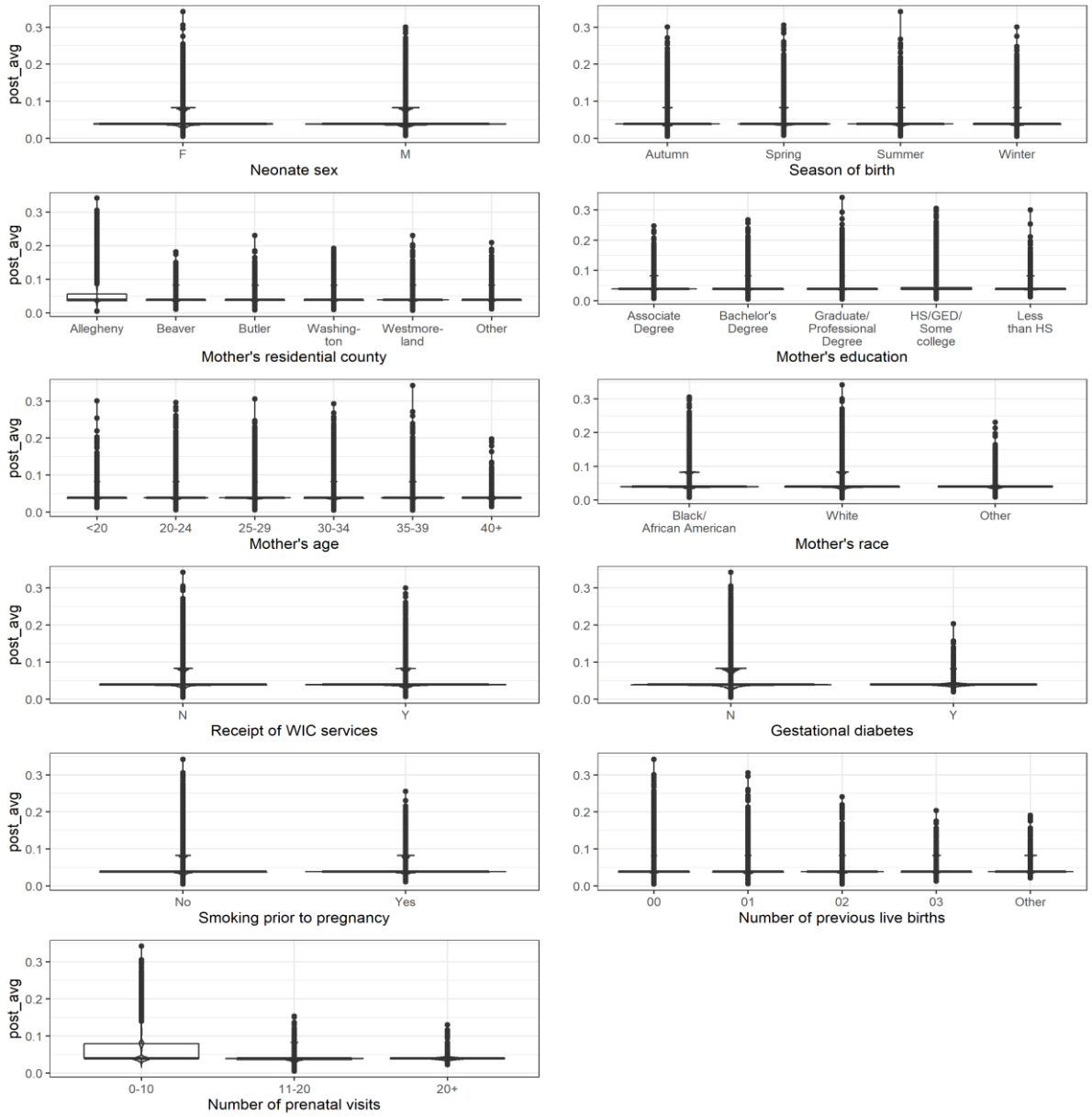

Figure 8 Distribution of posterior means for each categorical variable

Based on results from Figure 7, the categorical variable Number of Prenatal Visits (NOPV)'s 0-10 level is shown to be associated with highest posterior mean values, i.e., highest posterior average for the event probability is observed when NOPV = 0-10. Some association can also be seen when mother's residential county is Allegheny. These categorical variables are identified by Empirical Bayes as the important variables which are linked to preterm births. It would be interesting to see if the predictors identified as important by the classification models later in the analysis is synchronous to results from the Empirical Bayes exploration.

3.2.2 Performance of the models on the training and testing set

The down-sampled training data were used to build three machine learning classification models (Elastic Net, SVM and Random Forest) using a repeated cross validation with 5-folds and 3 repeats. Then, the trained models were used to evaluate the model performance on the testing set using the performance metrices AUC, Sensitivity and Specificity.

Results from the Elastic Net (GLMNET), SVM and Random Forest on the training set are shown in Figures 9 and 10. The visualization from these figures shows that Random Forest performed better than GLMNET and SVM in all three aspects of performance. Random Forest had the highest AUC value 0.83 compared to the AUC values for GLMNET and SVM, which were approximately 0.77 as seen in Figure 9. Then, Figure 10 shows that Random Forest had the highest sensitivity (0.767) compared to GLMNET (0.745) and SVM (0.754) and it also had the highest specificity (0.744) compared to GLMNET (0.715) and SVM (0.701) in the training dataset.

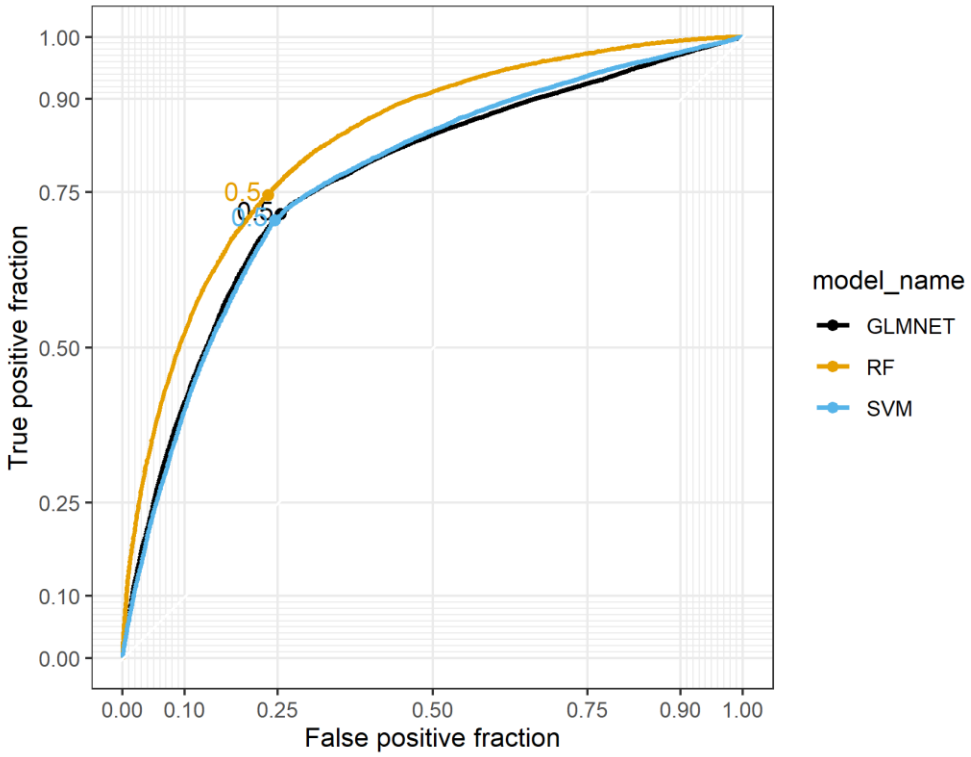

Figure 9 Model performance on the down-sampled training sets (ROC curves)

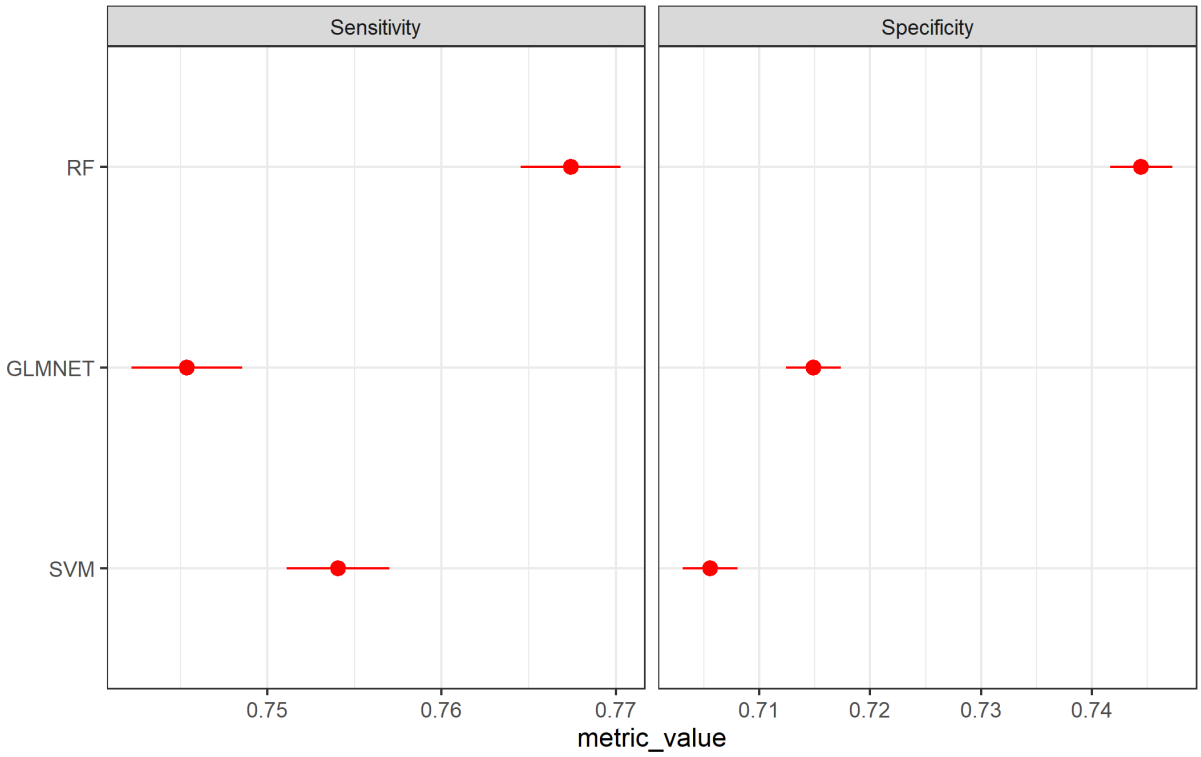

Figure 10 Model performance on the down-sampled training sets (Sensitivity and Specificity)

These three models were then used to predict on the testing dataset. The results of the performance metrics on the testing set are shown in Figures 11 and 12. Random Forest seems to perform the best compared to SVM and GLMNET in terms of AUC, Sensitivity, Specificity and Accuracy. Random Forest had the highest AUC value 0.83 compared to the AUC values for GLMNET and SVM as seen in Figure 11. Then, Figure 12 shows that Random Forest also performed the best in testing dataset in terms of accuracy, sensitivity and specificity.

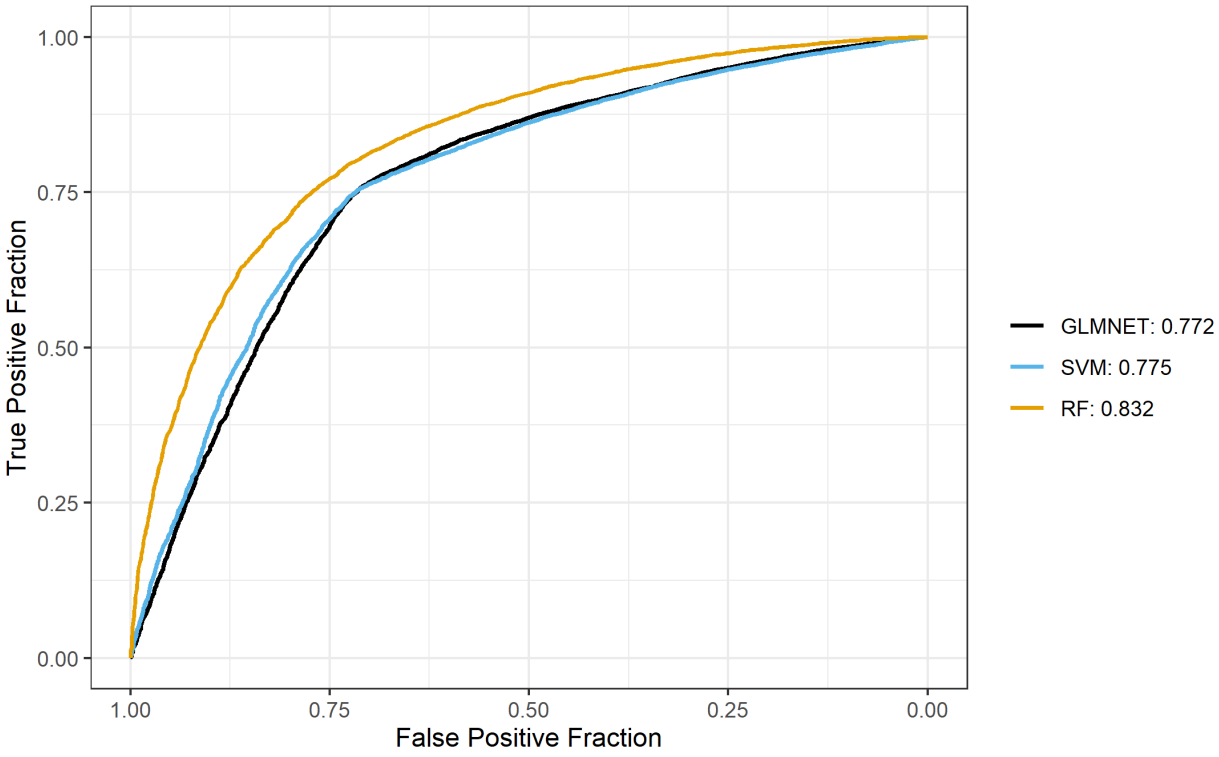

Figure 11 Model performance on the testing set (ROC curves)

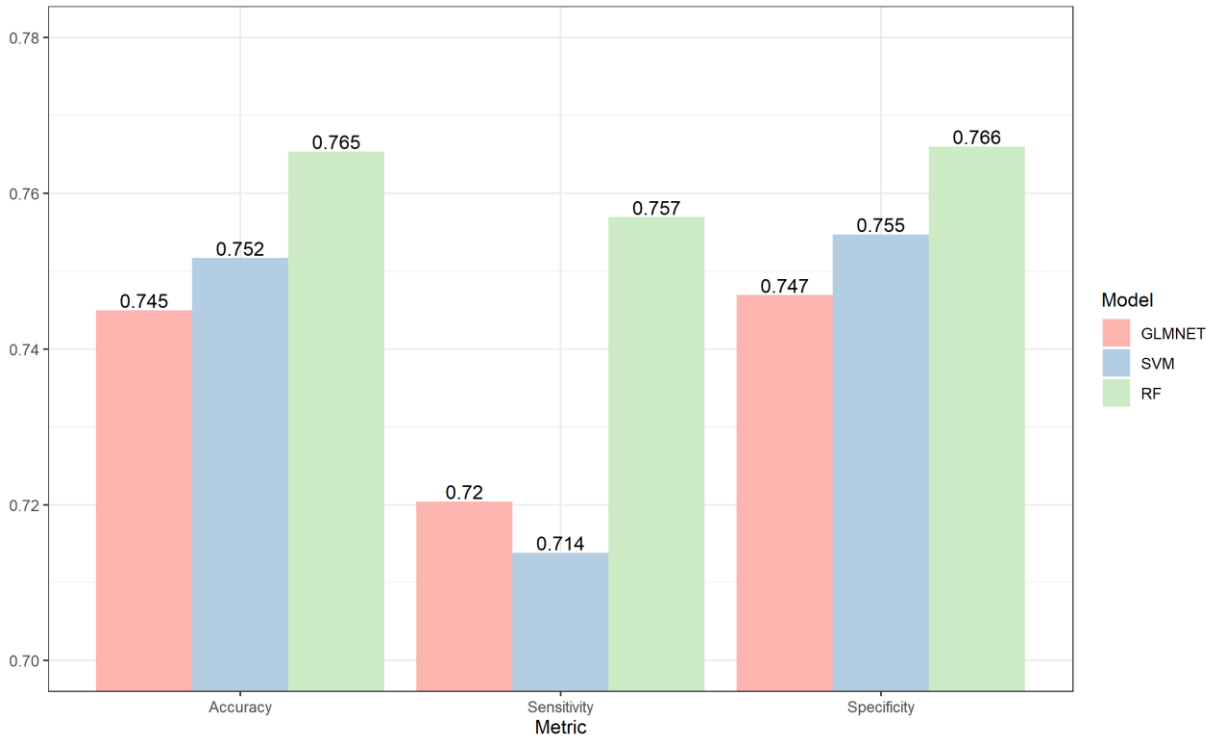

Figure 12 Model performance on the testing set (Accuracy, Sensitivity and Specificity)

To assess these important drivers of the response variable, i.e., preterm birth, a variable importance plot was created for each model which is shown in Figures 13-15. If multiple models rank same variables as important variables, it increases our confidence that those variables are indeed important to the prediction of the outcome. The plots in these figures reveal top ten predictors which are important to the prediction of preterm birth based on the models.

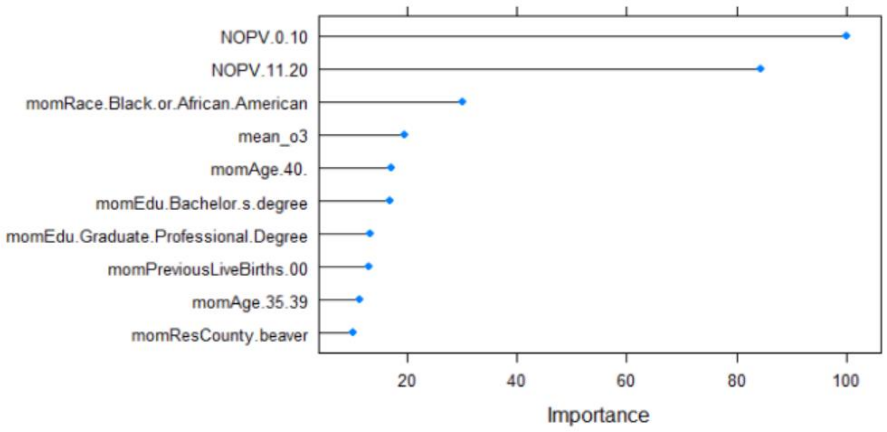

Figure 13 Variable Importance Plot for GLMNET

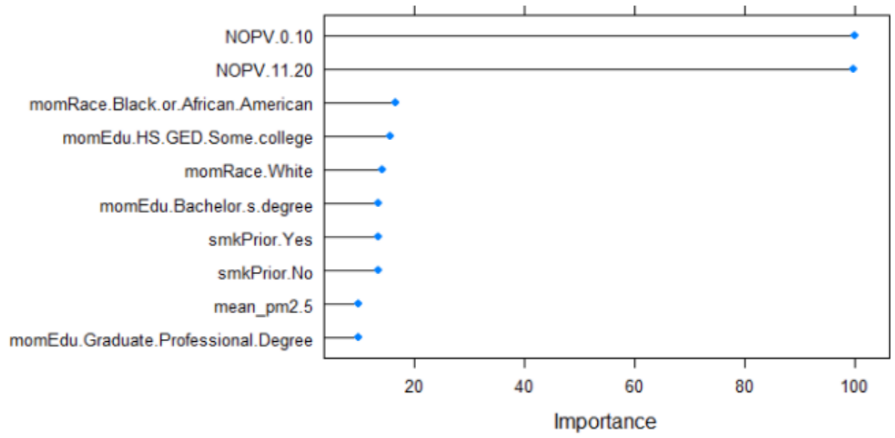

Figure 14 Variable Importance Plot for SVM

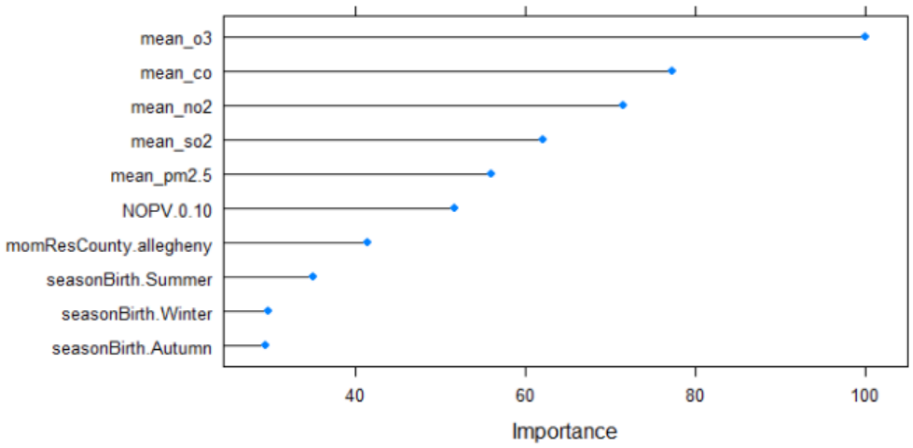

Figure 15 Variable Importance Plot for RF

GLMET and SVM both depict that number of prenatal visits, mother's race and education as some of the important predictors. Random Forest, on the other hand, shows that mean exposure to pollutants are the top important features. However, the caveat with variable importance plot for Random Forest is that categorical variables with many levels and continuous variables have much more plausible split points and since more of these splits will be utilized while building the decision trees, there is a higher chance that the variable would predict the outcome well. However, Random Forest model does show that number of prenatal visits is one of the top important features too, which is in line with the findings from the other two models.

4.0 Discussion

For this analysis, three classification models: Elastic Net (GLMNET), Support Vector machine (SVM) and Random Forest (RF) were evaluated using a 5-fold cross validation technique with 3 repeats to identify potential risk factors associated with preterm birth in southwestern PA. The pre-processed data contained 221,060 birth records from the year 2010 to 2020 with 16 predictor variables. Eleven of the predictors were categorical and 5 were continuous. After pre-processing, an Exploratory Data Analysis (EDA) was carried out to observe the distribution of both categorical and continuous variables by preterm births, which is explained in the Results section. To further explore the categorical data, an Empirical Bayes exploration was carried out. It was found that higher posterior means were associated with mothers who had fewer (0-10) number of prenatal visits, and mothers who resided in Allegheny County.

Next, a subsampling technique called down-sampling was utilized to handle the imbalance in the data, since about 93% of the data was labelled term birth and only 7% of the data was labelled preterm birth. Down-sampling was applied on the training data and three classification methods were used to train the data and build the models. The models utilized were Elastic Net (GLMNET), Support Vector Machine (SVM) and Random Forest. Random Forest model performed the best on the training data with an AUC of 0.830, sensitivity of 0.767 and specificity of 0.744. All three models were used to assess the model performance on the testing data. Random Forest performed the best on the testing data as well, with an AUC of 0.832 and sensitivity and specificity of 0.757 and 0.766 respectively.

To identify which variables could have significant roles in the prediction of preterm birth, variable importance plots for each of the three models were created. GLMNET and SVM identified

number of prenatal visits, mother's race and education as some of the important predictors. Random Forest showed that the mean exposure to pollutants are the top important features, with number of prenatal visits being one of the important predictors too. Random Forest also identified mother's residential county Allegheny as a top ten predictor. These results from Random Forest align with the results from Empirical Bayes exploration.

The results from this analysis are consistent with other literature. For instance, a study published on the American Journal of Obstetrics and Gynecology concluded that lack of prenatal care increased the relative risk for preterm birth by 2.8-fold in both African-American and white women (Vintzileos et al., 2002). Association between mother's education with preterm birth, as seen from the GLMNET and SVM models, is also consistent with findings from an Italian population-based study which reported that mothers with higher educational attainment had reduced odds (Odds Ratio = 0.81) of preterm births compared to less-educated mothers (Cantarutti et al., 2017). Similarly, a systematic review with meta-analysis reported that there is a positive association for the risk of preterm birth based on mother's race, with black women being at a higher risk of having preterm deliveries compared to non-black women (Oliveira et al., 2018). This is consistent with association that is seen between mother's race and preterm birth in this study from the GLMNET and SVM models. Lastly, even though Random Forest identified mean exposure to pollutants as some of the important predictors, further analysis is required to be certain of this finding because of the bias introduced by continuous variables in variable importance plot for Random Forest as discussed earlier in the Results section.

One of the limitations of this study is the interpretability of the models. Random Forest was selected as the best performing model on both the training and testing dataset; however, it is not possible to present the resulting model using a single decision tree. Therefore, even though the

Random Forest model performed the best, its interpretation is difficult (James et al., 2021). The other limitation is the bias in the variable importance plot for Random Forest. To overcome this drawback, one possible future iteration would be to identify response behavior associated with the top important features using a partial dependence plot. It helps us study predictive trends by understanding the marginal effect of any predictor variable on the predicted outcome variable (Boehmke et al., 2020).

Another possible future study based on this analysis would to be to further explore the categorical variable, Number of Prenatal Visits (NOPV), which was used to represent mother's access to prenatal care in the study. This variable was seen to be associated with preterm birth in all the three classification models and Empirical Bayes exploration. Identifying what it means when a mother does not have access to prenatal care in terms of socio-economic conditions, or demographic variables like mother's race or mother's residential county could help us further explore this question. We could try to see if the influence of environmental variables depends on socio-economic factors, while controlling for access to prenatal care. A better understanding of the predictors that might significantly contribute to preterm birth will help us target interventions to mothers and children in need in a timely manner, ultimately reducing the burden of preterm birth.

Appendix A Analysis Script: R Markdown

````
---
title: "Pre-Processing Code for Thesis"
output: html_document
---

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

```{r}
# install packages

install.packages("RMariaDB")
install.packages("tidyverse")
```

```{r}
# required libraries

library(RMariaDB)
library(tidyverse)
library(naniar)
```

```{r}
#connect MySQL Workbench to R

con <- dbConnect(RMariaDB::MariaDB(),
      default.file  = "C:/Users/sap196/.my.ini",
      group = "fracking-group")
dbListTables(con)
```

```{r}
# pull specific data from a specific table
statement <- 'select * from birth_data_Combined where momresstate = "Pennsylvania"'

# option 1
birth_data_Combined <- dbSendQuery(conn = con, statement = statement)
#dbFetch(birth_data_Combined)
````
````

```
# option 2
#dbGetQuery(conn = con, statement = statement)

## assign data to an object
birth_data_Combined <- dbGetQuery(conn = con, statement = statement)
str(birth_data_Combined)
head(birth_data_Combined)
```

```{r}
# create a new dataset called birth_data to explore the input and output variables of interest

birth_data <- birth_data_Combined
```

```{r}
write_rds(birth_data, 'birth_data.rds')
```

```{r}
# read in the data
birth_data <- read_rds('birth_data.rds')
```

```{r}
# filtering for multiple births
birth_data <- birth_data %>% filter(PLURAL==1)

# filtering for birth weight and gestational age
birth_data <- birth_data %>% filter(!LOP < 22)
birth_data <- birth_data %>% filter(!BWEIGHT < 500)


# selecting only the 8 counties
birth_data$momrescounty = tolower(birth_data$momrescounty)
birth_data <- birth_data %>% filter(momrescounty %in% c('allegheny', 'armstrong','beaver','butler','fayette','greene','washington','westmoreland'))

```

Preterm births

Any records with missing values for preterm births are excluded from the study.

```{r}
# create a new column with dichotomous outcome: preterm birth (yes/no)
```

```
birth_data <- birth_data %>% replace_with_na(replace = list(LOP = '99'))


birth_data <- birth_data %>% mutate(preterm_2 = ifelse(LOP < 37, "yes", "no"))
birth_data %>% count(preterm_2)
birth_data <- birth_data %>% drop_na(preterm_2)
```

Neonate sex

```{r}
# replace U's with NAs

table(birth_data$SEX)

birth_data <- birth_data %>% replace_with_na(replace = list(SEX = c("U")))
```

Season of birth

```{r}
# Spring (March–April–May), Summer (June–July–August), Autumn (September–October–November) and Winter (December–January–February)

birth_data$dob_month <- substr(birth_data$Child_DOB, 6, 7)

birth_data <- birth_data %>% mutate(season = case_when((dob_month == '12') | (dob_month == '01') | (dob_month == '02') ~ 'Winter',
                                    (dob_month == '03') | (dob_month == '04') | (dob_month == '05') ~ 'Spring',
                                    (dob_month == '06') | (dob_month == '07') | (dob_month == '08') ~ 'Summer',
                                    (dob_month == '09') | (dob_month == '10') | (dob_month == '11') ~ 'Autumn'))

```


Maternal Factors

Mother's education

```{r}
# replace 9s with NAs
birth_data <- birth_data %>% replace_with_na(replace = list(MOTHEDU=9))
```

```
birth_data <- birth_data %>% mutate(mom_edu=case_when(MOTHEDU == 1~ "8th grade or less",
                                    MOTHEDU == 2~ "9th - 12th grade; No diploma",
                                    MOTHEDU == 3~ "High school graduate or GED completed",
                                    MOTHEDU == 4~ "Some college credit, but not a degree",
                                    MOTHEDU == 5~ "Associate degree",
                                    MOTHEDU == 6~ "Bachelor's degree",
                                    MOTHEDU == 7~ "Master's degree",
                                    MOTHEDU == 8~ "Doctorate or Professional degree"))

# collapsing into fewer categories
birth_data <- birth_data %>% mutate(mom_edu_collapsed =case_when((MOTHEDU == 1) | (MOTHEDU == 2) ~ "Less than HS",
                                    (MOTHEDU == 3) | (MOTHEDU == 4)~ "HS/GED/Some college",
                                    MOTHEDU == 5~ "Associate degree",
                                    MOTHEDU == 6~ "Bachelor's degree",
                                    (MOTHEDU == 7) | (MOTHEDU == 8)~ "Graduate/Professional Degree"))

```

Mother's age

```{r}
# replace 99s with NAs
birth_data <- birth_data %>% replace_with_na(replace = list(MOTHAGE=99))

# categorize maternal age
birth_data <- birth_data %>% mutate(momAge = case_when((MOTHAGE < 20) ~ '<20',
                                    (MOTHAGE >= 20) & (MOTHAGE <= 24) ~ '20-24',
                                    (MOTHAGE >= 25) & (MOTHAGE <= 29) ~ '25-29',
                                    (MOTHAGE >= 30) & (MOTHAGE <= 34) ~ '30-34',
                                    (MOTHAGE >= 35) & (MOTHAGE <= 39) ~ '35-39',
                                    (MOTHAGE >= 40) ~ '40+'))
```

Mother's race

```{r}
table(birth_data$momRace)
```

```{r}
birth_data <- birth_data %>% mutate(momRace = case_when(MOTHRACE == 1 ~ "White",
```

```
                                    MOTHRACE == 2 ~ "Black or African American",
                                    MOTHRACE == 3 ~ "American Indian or Alaska Native",
                                    MOTHRACE == 4 | MOTHRACE == 5| MOTHRACE == 6 |
MOTHRACE == 7| MOTHRACE == 8| MOTHRACE == 9| MOTHRACE == 10| MOTHRACE
== 11 | MOTHRACE == 12 | MOTHRACE == 13 | MOTHRACE == 14 ~ "Asian/PI",
                                    MOTHRACE == 15 ~ "Other"))

```

Pre-pregnancy body mass index

```{r}
birth_data$MPPWGT <- as.numeric(birth_data$MPPWGT)
birth_data$momhtFeet <- as.numeric(birth_data$momhtFeet)
birth_data$momhtInches <- as.numeric(birth_data$momhtInches)

birth_data <- birth_data %>% replace_with_na(replace = list(momhtFeet=9, momhtInches=99,
MPPWGT=999))

# change height to meters and weight to kg
birth_data$momhtMeters <- (birth_data$momhtFeet * 0.3048) + (birth_data$momhtInches *
0.0254)
birth_data$momweightKg <- birth_data$MPPWGT * 0.453592

birth_data$BMI <- ((birth_data$momweightKg)/(birth_data$momhtMeters^2))

```

BMI class

I categorized BMI into following classes:

1. BMI < 18. -> 'Underweight'
2. BMI >= 18.5 & BMI <= 24.9 -> 'Normal'
3. BMI >= 25.0 & BMI <= 29.9 -> 'Overweight'
4. BMI >= 30.0 -> 'Obese'

```{r}
birth_data <- birth_data %>% mutate(BMI_class = case_when((BMI < 18.5)  ~ 'Underweight',
                                    (BMI >= 18.5) & (BMI <= 24.9) ~ 'Normal',
                                    (BMI >= 25.0) & (BMI <= 29.9) ~ 'Overweight',
                                    (BMI >= 30.0) ~ 'Obese'))
```

Smoking status during pregnancy

Number of cigarettes smoked three months prior, and during the first, second, and third trimesters. I dichotomized the smoking variables as yes/no for each of these categories.

```{r}
# no. of cigarettes smoked three months prior

# replace 99s with NAs
birth_data <- birth_data %>% replace_with_na(replace = list(SMKPR=99))

# categorize smoking 3 months prior as Yes/No

birth_data <- birth_data %>% mutate(SMKPR_YN = case_when(SMKPR == 0 ~ 'No',
                                SMKPR > 0 ~ 'Yes'))

```

```{r}
# no. of cigarettes smoked first three months

# replace 99s with NAs
birth_data <- birth_data %>% replace_with_na(replace = list(SMKFTM=99))

# categorize smoking in 1st trimester as Yes/No

birth_data <- birth_data %>% mutate(SMKFTM_YN = case_when(SMKFTM == 0 ~ 'No',
                                SMKFTM > 0 ~ 'Yes'))
```

```{r}
# no. of cigarettes smoked second three months

# replace 99s with NAs
birth_data <- birth_data %>% replace_with_na(replace = list(SMKSTM=99))

# categorize smoking in 2nd trimester as Yes/No

birth_data <- birth_data %>% mutate(SMKSTM_YN = case_when(SMKSTM == 0 ~ 'No',
                                SMKSTM > 0 ~ 'Yes'))

```

```{r}
# no. of cigarettes smoked last three months

# replace 99s with NAs
```

```
birth_data <- birth_data %>% replace_with_na(replace = list(SMKLTM=99)) # there were 3053 NAs

birth_data %>% ggplot(aes(x=SMKLTM)) + geom_bar()

# categorize smoking in 3rd trimester as Yes/No

birth_data <- birth_data %>% mutate(SMKLTM_YN = case_when(SMKLTM == 0 ~ 'No',
                                    SMKLTM > 0 ~ 'Yes'))
```

Checking correlation between the smoking variable

```{r}
birth_data %>% count(SMKFTM_YN, SMKSTM_YN, SMKLTM_YN) %>% drop_na()
```

Looks like the variables are related, since the Yes results and the No results are lined up.

```{r}
# create categories based on unique combinations of smoking Yes/No

birth_data <- birth_data %>% unite(SMK_comb, c(SMKFTM_YN, SMKSTM_YN, SMKLTM_YN), sep = ",", remove=FALSE)
```

Total number of prenatal visits (NOPV)

```{r}
# replace 99s and 88s with NAs
birth_data <- birth_data %>% replace_with_na(replace = list(NOPV=c(99,88)))
```

Receipt of WIC services (a surrogate for low family socioeconomic status)

```{r}
# replace X's or U's with NAs
birth_data <- birth_data %>% replace_with_na(replace = list(MWIC = c("X","U")))
```

Categorize NOPV based on literature

"Based on a prior study by Buekens that found the median number of PNV in the United States was 11"

(Source: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4767570/ and Buekens P, Kotelchuck M, Blondel B, Kristensen FB, Chen JH, Masuy-Stroobant G. A comparison of prenatal care use in the United States and Europe. American Journal of Public Health. 1993;83(1):31–6.)

In this data set, mean Number of Prenatal visits is 11.95 ~ 12 and the median is 13.

```{r}
birth_data %>% select(NOPV) %>% summarize(mean=mean(NOPV, na.rm=T),
                                median=median(NOPV, na.rm=T))
```

```{r}
# creating categories
birth_data <- mutate(birth_data, NOPV_c = case_when(NOPV <= 10 ~ '0-10',
                                (NOPV >= 11) & (NOPV <= 20) ~ '11-20',
                                (NOPV >= 21)~ '20+'))

```

```{r}
# calculate the time the child was in womb (Start date and end date)

birth_data$wombStart <- birth_data$Child_DOB - birth_data$LOP*7
```

```{r}
# select required columns

birth_data <- birth_data %>% select(Birth_ID, Child_DOB, SEX, momrescounty, MWIC, R2,LBIRTH, DERIVED_lat, DERIVED_long, preterm_2, season, mom_edu_collapsed, momAge, momRace, BMI_class, SMKPR_YN, SMK_comb, NOPV_c, wombStart)
```

```{r}
glimpse(birth_data)
```

```{r}
# save it as a rds file
write_rds(birth_data, 'birth_data_1.rds')
```
---
title: "Air Data Code for Thesis"
output: html_document

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

Read in finalized birth data

```{r}
library(tidyverse)
library(readxl)
library(geosphere)
```

Read in files from 2009-2020 for ozone

```{r message=FALSE, warning=FALSE, comment=""}

rm(list=ls())

# set the working directory to where the excel files are

setwd("C:/Users/sap196/Desktop/EPA Data/Ozone_excel_files")


# list files with .csv as the extension
my_files <- list.files(pattern="*.csv")

ozone <- lapply(my_files, function(i){
  x = read_csv(i)
  x$file=i
  x
})


ozone =do.call("rbind.data.frame",ozone)

ozone_data <- ozone

```

```{r}
ozone_data <- ozone_data %>% filter(`State Name` == "Pennsylvania")
```

```{r}
```

```
write_rds(ozone_data, 'ozone.rds')
```

SO2
Read in files from 2009-2020 for SO2

```{r message=FALSE, warning=FALSE, comment=""}

rm(list=ls())

# set the working directory to where the excel files are

setwd("C:/Users/sap196/Desktop/MS 2022/EPA Data/SO2_excel_files")


# list files with .csv as the extension
my_files <- list.files(pattern="*.csv")

so2 <- lapply(my_files, function(i){
  x = read_csv(i)
  x$file=i
  x
})


#comb[[4]]
so2 =do.call("rbind.data.frame",so2)

so2_data <- so2

```

```{r}
so2_data <- so2_data %>% filter(`State Name` == "Pennsylvania")
```

```{r}
write_rds(so2_data, 'so2.rds')
```

NO2
Read in files from 2009-2020 for NO2

```{r message=FALSE, warning=FALSE, comment=""}

rm(list=ls())
```

set the working directory to where your excel files are

setwd("C:/Users/sap196/Desktop/MS 2022/EPA Data/NO2_excel_files")

# list files with .csv as the extension
my_files <- list.files(pattern="*.csv")

no2 <- lapply(my_files, function(i){
  x = read_csv(i)
  x$file=i
  x
})

#comb[[4]]
no2 =do.call("rbind.data.frame",no2)

no2_data <- no2

```

```{r}
no2_data <- no2_data %>% filter(`State Name` == "Pennsylvania")
```

```{r}
write_rds(no2_data, 'no2.rds')
```

CO
Read in files from 2009-2020 for CO

```{r message=FALSE, warning=FALSE, comment=""}

rm(list=ls())

# set the working directory to where your excel files are

setwd("C:/Users/sap196/Desktop/MS 2022/EPA Data/CO_excel_files")


# list files with .csv as the extension
my_files <- list.files(pattern="*.csv")

```
co <- lapply(my_files, function(i){
  x = read_csv(i)
  x$file=i
  x
})

#comb[[4]]
co =do.call("rbind.data.frame",co)

co_data <- co

```

```{r}
co_data <- co_data %>% filter(`State Name` == "Pennsylvania")
```

```{r}
write_rds(co_data, 'co.rds')
```

PM2.5
Read in files from 2009-2020 for PM2.5

```{r message=FALSE, warning=FALSE, comment=""}

rm(list=ls())

# set the working directory to where your excel files are

setwd("C:/Users/sap196/Desktop/MS 2022/EPA Data/PM2.5_excel_files")


# list files with .csv as the extension
my_files <- list.files(pattern="*.csv")

pm2.5 <- lapply(my_files, function(i){
  x = read_csv(i)
  x$file=i
  x
})


#comb[[4]]
pm2.5 =do.call("rbind.data.frame",pm2.5)
```

pm2.5_data <- pm2.5

```

```{r}
pm2.5_data <- pm2.5_data %>% filter(`State Name` == "Pennsylvania")
```

```{r}
write_rds(pm2.5_data, 'pm2.5.rds')
```

```{r}
birth_data <- read_rds('birth_data_1.rds')
glimpse(birth_data)
```

Subset to fewer columns

```{r}
birth_data_epa <- birth_data %>% select(Birth_ID, Child_DOB, DERIVED_lat, DERIVED_long, wombStart)

### only keep the birth data with complete location coordinates
birth_data_epa <- birth_data_epa %>%
  filter(!is.na(DERIVED_lat)) %>%
  filter(!is.na(DERIVED_long))

sum(is.na(birth_data_epa$DERIVED_lat))
sum(is.na(birth_data_epa$DERIVED_long))
```

Select required columns only from EPA data

- State Code, County Code, Site Num
- Parameter Code, POC
- Latitude, Longitude
- Parameter name
- Date Local
- Unit of Measure (ppm)
- Arithmetic Mean
- AQI
- Local Site Name
- Address
- State Name
```

- County name
- City Name
- CBSA name
- Date of Last Change

Excluded for now:

- Datum
- Sample Duration (8-HR RUN AVG BEGIN HOUR)
- Pollutant Standard
- EVENT TYPE
- Obs Count, Obs Percent
- Other columns

Part 1: Ozone

```{r}
ozone_data <- read_rds('ozone.rds')

ozone_data <- ozone_data %>% filter(`County Name` %in% c("Allegheny", "Armstrong","Beaver","Butler","Fayette","Greene","Washington","Westmoreland"))

ozone_data <- ozone_data %>% select(1,2,3,4,5,6,7,9,12,13,17,20,23,24,25,26,27,28,29)

# Year of data collection and year last updated

ozone_data$YearCollect <- substr(ozone_data$`Date Local`, 1,4)
ozone_data$YearLast <- substr(ozone_data$`Date of Last Change`, 1,4)

ozone_data <- ozone_data %>% select(c(2,3,6,7,8,9,11,12))
```

```{r}
# Approach - I (for-loop)
# For O3

# start.time <- Sys.time()
stat_ozone <- vector(mode = 'list', length = nrow(birth_data_epa))

for(i in 1:nrow(birth_data_epa)){
  a <- ozone_data %>% filter(`Date Local` >= birth_data_epa[i,]$wombStart & `Date Local` <= birth_data_epa[i,]$Child_DOB)

  a_site <- a %>% distinct(`Site Num`, Latitude, Longitude)

  for (j in 1:nrow(a_site)){
```

```
    a_site$childID <- birth_data_epa[i,]$Birth_ID
    a_site$dist[j] <- distm(c(birth_data_epa[i,]$DERIVED_long, birth_data_epa[i,]$DERIVED_lat), c(a_site[j,]$Longitude, a_site[j,]$Latitude), fun=distHaversine)
  }
  a_site <- a_site[which.min(a_site$dist),]

  stat_ozone[[i]] <- a %>% filter(`Site Num` == a_site$`Site Num` & Latitude == a_site$Latitude & Longitude == a_site$Longitude) %>% summarise(n= n(),mean = mean(`Arithmetic Mean`), md_o3 = median(`Arithmetic Mean`), var_o3=var(`Arithmetic Mean`))

  stat_ozone[[i]]$site_dist <- a_site$dist
  stat_ozone[[i]]$Birth_ID <- a_site$childID
}

df_ozone <- do.call(rbind.data.frame, stat_ozone)
write_rds(df_ozone, 'df_ozone_SP_1.rds')

# end.time <- Sys.time()
# time.taken.o3 <- round(end.time - start.time,2)
# time.taken.o3
```

Part 2: SO2

```{r}
so2_data <- read_rds('so2.rds')

so2_data <- so2_data %>% filter(`County Name` %in% c("Allegheny", "Armstrong","Beaver","Butler","Fayette","Greene","Washington","Westmoreland"))

so2_data <- so2_data %>% select(1,2,3,4,5,6,7,9,12,13,17,20,23,24,25,26,27,28,29)

# Year of data collection and year last updated

so2_data$YearCollect <- substr(so2_data$`Date Local`, 1,4)
so2_data$YearLast <- substr(so2_data$`Date of Last Change`, 1,4)

so2_data <- so2_data %>% select(c(2,3,6,7,8,9,11,12))
```

```{r}
# Approach - I (for-loop)
# For so2
```

```
#start.time <- Sys.time()
stat_so2 <- vector(mode = 'list', length = nrow(birth_data_epa))

for(i in 1:nrow(birth_data_epa)){
  a <- so2_data %>% filter(`Date Local` >= birth_data_epa[i,]$wombStart & `Date Local` <= birth_data_epa[i,]$Child_DOB)

  a_site <- a %>% distinct(`Site Num`, Latitude, Longitude)

  for (j in 1:nrow(a_site)){
    a_site$childID <- birth_data_epa[i,]$Birth_ID
    a_site$dist[j] <- distm(c(birth_data_epa[i,]$DERIVED_long, birth_data_epa[i,]$DERIVED_lat), c(a_site[j,]$Longitude, a_site[j,]$Latitude), fun=distHaversine)
  }
  a_site <- a_site[which.min(a_site$dist),]

  stat_so2[[i]] <- a %>% filter(`Site Num` == a_site$`Site Num` & Latitude == a_site$Latitude & Longitude == a_site$Longitude) %>% summarise(n= n(),mean = mean(`Arithmetic Mean`), md_o3 = median(`Arithmetic Mean`), var_o3=var(`Arithmetic Mean`))

  stat_so2[[i]]$site_dist <- a_site$dist
  stat_so2[[i]]$Birth_ID <- a_site$childID
}

df_so2 <- do.call(rbind.data.frame, stat_so2)
write_rds(df_so2, 'df_so2_SP_1.rds')

# end.time <- Sys.time()
# time.taken.so2 <- round(end.time - start.time,2)
# time.taken.so2
```

Part 3: NO2

```{r}
no2_data <- read_rds('no2.rds')

no2_data <- no2_data %>% filter(`County Name` %in% c("Allegheny", "Armstrong","Beaver","Butler","Fayette","Greene","Washington","Westmoreland"))

no2_data <- no2_data %>% select(1,2,3,4,5,6,7,9,12,13,17,20,23,24,25,26,27,28,29)

# Year of data collection and year last updated

no2_data$YearCollect <- substr(no2_data$`Date Local`, 1,4)
```

```
no2_data$YearLast <- substr(no2_data$`Date of Last Change`, 1,4)

#table(no2_data$YearCollect, no2_data$YearLast)
no2_data <- no2_data %>% select(c(2,3,6,7,8,9,11,12))
```

```{r}
# Approach - I (for-loop)
# For no2

# start.time <- Sys.time()
stat_no2 <- vector(mode = 'list', length = nrow(birth_data_epa))

for(i in 1:nrow(birth_data_epa)){
  a <- no2_data %>% filter(`Date Local` >= birth_data_epa[i,]$wombStart & `Date Local` <= birth_data_epa[i,]$Child_DOB)

  a_site <- a %>% distinct(`Site Num`, Latitude, Longitude)

  for (j in 1:nrow(a_site)){
    a_site$childID <- birth_data_epa[i,]$Birth_ID
    a_site$dist[j] <- distm(c(birth_data_epa[i,]$DERIVED_long, birth_data_epa[i,]$DERIVED_lat), c(a_site[j,]$Longitude, a_site[j,]$Latitude), fun=distHaversine)
  }
  a_site <- a_site[which.min(a_site$dist),]

  stat_no2[[i]] <- a %>% filter(`Site Num` == a_site$`Site Num` & Latitude == a_site$Latitude & Longitude == a_site$Longitude) %>% summarise(n= n(),mean_no2 = mean(`Arithmetic Mean`), md_no2 = median(`Arithmetic Mean`), var_no2=var(`Arithmetic Mean`))

  stat_no2[[i]]$site_dist <- a_site$dist
  stat_no2[[i]]$Birth_ID <- a_site$childID

  # median_no2[[i]]
  # var_no2[[i]]
}

df_no2 <- do.call(rbind.data.frame, stat_no2)
write_rds(df_no2, 'df_no2_SP.rds')

# end.time <- Sys.time()
# time.taken.no2 <- round(end.time - start.time,2)
# time.taken.no2
```

Part 4: CO

```{r}
co_data <- read_rds('co.rds')

co_data <- co_data %>% filter(`County Name` %in% c("Allegheny", "Armstrong","Beaver","Butler","Fayette","Greene","Washington","Westmoreland"))

co_data <- co_data %>% select(1,2,3,4,5,6,7,9,12,13,17,20,23,24,25,26,27,28,29)

# Year of data collection and year last updated

co_data$YearCollect <- substr(co_data$`Date Local`, 1,4)
co_data$YearLast <- substr(co_data$`Date of Last Change`, 1,4)

#table(co_data$YearCollect, co_data$YearLast)
co_data <- co_data %>% select(c(2,3,6,7,8,9,11,12))
```

```{r}
# Approach - I (for-loop)
# For co

# start.time <- Sys.time()
stat_co <- vector(mode = 'list', length = nrow(birth_data_epa))

for(i in 1:nrow(birth_data_epa)){
  a <- co_data %>% filter(`Date Local` >= birth_data_epa[i,]$wombStart & `Date Local` <= birth_data_epa[i,]$Child_DOB)

  a_site <- a %>% distinct(`Site Num`, Latitude, Longitude)

  for (j in 1:nrow(a_site)){
    a_site$childID <- birth_data_epa[i,]$Birth_ID
    a_site$dist[j] <- distm(c(birth_data_epa[i,]$DERIVED_long, birth_data_epa[i,]$DERIVED_lat), c(a_site[j,]$Longitude, a_site[j,]$Latitude), fun=distHaversine)
  }
  a_site <- a_site[which.min(a_site$dist),]

  stat_co[[i]] <- a %>% filter(`Site Num` == a_site$`Site Num` & Latitude == a_site$Latitude & Longitude == a_site$Longitude) %>% summarise(n= n(),mean_co = mean(`Arithmetic Mean`), md_co = median(`Arithmetic Mean`), var_co=var(`Arithmetic Mean`))

  stat_co[[i]]$site_dist <- a_site$dist
```

```
  stat_co[[i]]$Birth_ID <- a_site$childID


}

df_co <- do.call(rbind.data.frame, stat_co)
write_rds(df_co, 'df_co_SP.rds')

# end.time <- Sys.time()
# time.taken.co <- round(end.time - start.time,2)
# time.taken.co
```

Part 5: PM2.5

```{r}
pm2.5_data <- read_rds('pm2.5.rds')

pm2.5_data <- pm2.5_data %>% filter(`County Name` %in% c("Allegheny", "Armstrong","Beaver","Butler","Fayette","Greene","Washington","Westmoreland"))

pm2.5_data <- pm2.5_data %>% select(1,2,3,4,5,6,7,9,12,13,17,20,23,24,25,26,27,28,29)

# Year of data collection and year last updated

pm2.5_data$YearCollect <- substr(pm2.5_data$`Date Local`, 1,4)
pm2.5_data$YearLast <- substr(pm2.5_data$`Date of Last Change`, 1,4)

#table(pm2.5_data$YearCollect, pm2.5_data$YearLast)
pm2.5_data <- pm2.5_data %>% select(c(2,3,6,7,8,9,11,12))
```

```{r}
# Approach - I (for-loop)
# For pm2.5

# start.time <- Sys.time()
stat_pm2.5 <- vector(mode = 'list', length = nrow(birth_data_epa))

for(i in 1:nrow(birth_data_epa)){
  a <- pm2.5_data %>% filter(`Date Local` >= birth_data_epa[i,]$wombStart & `Date Local` <= birth_data_epa[i,]$Child_DOB)

  a_site <- a %>% distinct(`Site Num`, Latitude, Longitude)

  for (j in 1:nrow(a_site)){
```

```
    a_site$childID <- birth_data_epa[i,]$Birth_ID
    a_site$dist[j] <- distm(c(birth_data_epa[i,]$DERIVED_long, birth_data_epa[i,]$DERIVED_lat), c(a_site[j,]$Longitude, a_site[j,]$Latitude), fun=distHaversine)
  }
  a_site <- a_site[which.min(a_site$dist),]

  stat_pm2.5[[i]] <- a %>% filter(`Site Num` == a_site$`Site Num` & Latitude == a_site$Latitude & Longitude == a_site$Longitude) %>% summarise(n= n(),mean_pm2.5 = mean(`Arithmetic Mean`), md_pm2.5 = median(`Arithmetic Mean`), var_pm2.5=var(`Arithmetic Mean`))

  stat_pm2.5[[i]]$site_dist <- a_site$dist
  stat_pm2.5[[i]]$Birth_ID <- a_site$childID


}

df_pm2.5 <- do.call(rbind.data.frame, stat_pm2.5)
write_rds(df_pm2.5, 'df_pm2.5_SP.rds')

# end.time <- Sys.time()
# time.taken.co <- round(end.time - start.time,2)
# time.taken.co
```

Plot maps

```{r}
o3_map <- ozone_data %>% select(Latitude, Longitude, `Parameter Name`)
so2_map <- so2_data %>% select(Latitude, Longitude, `Parameter Name`)
no2_map <- no2_data %>% select(Latitude, Longitude, `Parameter Name`)
co_map <- co_data %>% select(Latitude, Longitude, `Parameter Name`)
pm2.5_map <- pm2.5_data %>% select(Latitude, Longitude, `Parameter Name`)

map <- rbind(o3_map, so2_map, no2_map, co_map, pm2.5_map)
```

```{r}
table(map$`Parameter Name`)
```

```{r}
facet_labels <- c(
  `Carbon monoxide` = "Carbon monoxide",
  `Nitrogen dioxide (NO2)` = "Nitrogen dioxide",
  `Ozone` = "Ozone",
```

```
  `PM2.5 - Local Conditions` = "Particulate Matter 2.5",
  `Sulfur dioxide` = "Sulfur dioxide"
)

map_plot <- qmplot(Longitude, Latitude, data = map, colour = `Parameter Name`, size = I(1), darken = .05)+facet_wrap(~`Parameter Name`, labeller = as_labeller(facet_labels))+ theme(panel.border = element_blank(),
      panel.background = element_blank(),
      panel.grid = element_blank(),
      panel.spacing.x = unit(1,"line"),
      panel.spacing.y = unit(1,"line"))
```

```{r}
map_plot
```

```{r}
ggsave("map_plot.png")  ## save plot
```
---
title: "Merge and EDA for Thesis"
output: html_document

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

```{r}
library(tidyverse)
```

Merging birth data to pollutants data

```{r}
birth_data <- read_rds("birth_data_1.rds")
```

```{r}
birth_data <- birth_data %>% drop_na(DERIVED_lat, DERIVED_long)

ozone_conc <- read_rds('df_ozone_SP_1.rds')
so2_conc <- read_rds('df_so2_SP_1.rds')
no2_conc <- read_rds('df_no2_SP.rds')
co_conc <- read_rds('df_co_SP.rds')
```

```
pm2.5_conc <- read_rds('df_pm2.5_SP.rds')

ozone_conc <- rename(ozone_conc, mean_o3 = mean, n_o3 = n, site_dist_o3 = site_dist)
so2_conc <- rename(so2_conc, c(mean_so2 = mean, md_so2 = md_o3, var_so2 = var_o3, n_so2
= n, site_dist_so2 = site_dist))
no2_conc <- rename(no2_conc, n_no2 = n, site_dist_no2 = site_dist)
co_conc <- rename(co_conc, n_co = n, site_dist_co = site_dist)
pm2.5_conc <- rename(pm2.5_conc, n_pm2.5 = n, site_dist_pm2.5 = site_dist)


birth_data_w_o3 <- merge(birth_data, ozone_conc, by="Birth_ID")
birth_data_w_so2 <- merge(birth_data, so2_conc, by="Birth_ID")
birth_data_w_no2 <- merge(birth_data, no2_conc, by="Birth_ID")
birth_data_w_co <- merge(birth_data, co_conc, by="Birth_ID")
birth_data_w_pm2.5 <- merge(birth_data, pm2.5_conc, by="Birth_ID")

# one big dataset

birth_data_final <- merge(birth_data, ozone_conc, by="Birth_ID")
birth_data_final <- merge(birth_data_final, so2_conc, by="Birth_ID")
birth_data_final <- merge(birth_data_final, no2_conc, by="Birth_ID")
birth_data_final <- merge(birth_data_final, co_conc, by="Birth_ID")
birth_data_final <- merge(birth_data_final, pm2.5_conc, by="Birth_ID")

write_rds(birth_data_final, 'birth_data_final.rds')
```


lumping together variables with low frequency

```{r}
df <- read_rds('birth_data_final.rds')
```

```{r}
table(df$momRace)
```

```{r}
### lump together all levels with less than 5% frequency
df <- df %>%
  mutate(SMK_comb = forcats::fct_lump_prop(SMK_comb, 0.05))

df <- df %>%
  mutate(momRace = forcats::fct_lump_prop(momRace, 0.05))

### another categorical with levels with very low proportions
```

```
df <- df %>%
  mutate(LBIRTH = forcats::fct_lump_prop(LBIRTH, 0.05))

### a few low proportion levels
df <- df %>%
  mutate(momrescounty = forcats::fct_lump_prop(momrescounty, 0.05))
```

```{r}
### for SMK_comb
df %>%
  ggplot(mapping = aes(x = SMK_comb)) +
  geom_bar(mapping = aes(y = stat(prop),
                         group = 1)) +
  geom_text(stat = 'count',
            mapping = aes(y = after_stat(count / sum(count)),
                          label = after_stat( signif(count / sum(count), 2))),
            nudge_y = 0.03, color = 'red') +
  coord_flip() +
  theme_bw()
```

```{r}
### other smoker variable
df %>%
  ggplot(mapping = aes(y = SMKPR_YN)) +
  geom_bar() +
  theme_bw()
```

So, SMK_comb is effectively redundant with SMKPR_YN due to the very low frequency for some of the levels.

```{r}
# dropping unwanted columns

df <- df %>% dplyr::select(-Birth_ID, -DERIVED_lat, -DERIVED_long, -SMK_comb, -wombStart, -n_o3, -var_o3, -site_dist_o3,
                  -n_so2, -var_so2, -site_dist_so2,
                  -n_no2, -var_no2, -site_dist_no2,
                  -n_co, -var_co, -site_dist_co,
                  -n_pm2.5, -var_pm2.5, -site_dist_pm2.5)
df
```

```{r}
# explore correlation between mean and median exposures
df %>%
  select(c(14:23)) %>%
  cor() %>%
  corrplot::corrplot(method = 'square', type = 'upper',
                     order = 'hclust', hclust.method = 'ward.D2')
```

```{r}
 # droping median

df <- df %>% select(-md_o3, -md_so2, -md_no2, -md_co, -md_pm2.5)
```

```{r fig.height=8, fig.width=12}
# missing data visualization
# using naniar

library(visdat)
vis_dat(df, warn_large_data = FALSE) + theme(axis.text.x = element_text(size = 8, angle = 90))
```
```{r}
df <- df %>% rename("NOPV" = "NOPV_c",
                    "smkPrior" ="SMKPR_YN",
                    "momEdu" = "mom_edu_collapsed",
                    "neonateSex" = "SEX",
                    "momResCounty" = "momrescounty",
                    "momWIC" = "MWIC",
                    "momGestDiabetes" = "R2",
                    "momPreviousLiveBirths" = "LBIRTH",
                    "BMI" = "BMI_class",
                    "seasonBirth" = "season"
                    )
```

```{r fig.height=8, fig.width=12}
vis_miss(df %>% select(-c(14:18)), sort_miss=TRUE, warn_large_data = FALSE)+
theme(axis.text.x = element_text(size = 12, angle = 90))
```

```{r}
# distribution for BMI missing vs non-missing

df %>% filter(is.na(BMI)) %>%
```

```
  ggplot(aes(x=preterm_2)) + geom_bar() + geom_text(stat='count', aes(label=paste0(c(..count..),
"(", scales::percent(..count../sum(..count..)), ")")), color = "black", size = 3.5) +
  theme(axis.text.x = element_text(angle = 0))

df %>% filter(!is.na(BMI)) %>%
  ggplot(aes(x=preterm_2)) + geom_bar() + geom_text(stat='count', aes(label=paste0(c(..count..),
"(", scales::percent(..count../sum(..count..)), ")")), color = "black", size = 3.5) +
  theme(axis.text.x = element_text(angle = 0))
```

```{r}
df <- df %>%  mutate_if(is.character, as.factor)

glimpse(df)

df$preterm_2 <- relevel(df$preterm_2, "yes")
```

```{r}
df <- df %>% select(-BMI_class)
df <- df %>% drop_na()
```

```{r}
write_rds(df, 'df.rds')
```

```{r}
table(df$momResCounty)
```

Descriptive statistics table for variables

Categorical
```{r}
library(reshape2)
library(dplyr)

df_desc <-
melt(df,measure.vars=c("neonateSex","momResCounty","momWIC","momGestDiabetes","mo
mPreviousLiveBirths","seasonBirth","momAge","momEdu","smkPrior","NOPV","momRace"))

res <- df_desc %>%
  group_by(variable, value, preterm_2) %>% summarize(n = n()) %>%
  mutate(freq =n / sum(n))
res
```

```

```{r}
#make an 'export' variable
res$export <- with(res, sprintf("%i (%.1f%%)", n, freq*100))

#reshape again
output <- dcast(variable+value~preterm_2, value.var="export", data=res, fill="missing") #use drop=F to prevent silent missings
#'silent missings'
output$variable <- as.character(output$variable)
#make 'empty lines'
empties <- data.frame(variable=unique(output$variable), stringsAsFactors=F)
empties[,colnames(output)[-1]] <- ""

#bind them together
output2 <- rbind(empties,output)
output2 <- output2[order(output2$variable,output2$value),]

#optional: 'remove' variable if value present

output2$variable[output2$value!=""] <- ""
```

```{r}
output2
```

```{r}
# Write this table to a comma separated .txt file:
write.table(output2, file = "desc_stat.txt", sep = ",", quote = FALSE, row.names = F)
```

Continuous variables

```{r}

df_desc_cont <- melt(df,measure.vars=c("mean_co","mean_no2","mean_o3","mean_pm2.5","mean_so2"))

res_cont <- df_desc_cont %>%
  group_by(variable, preterm_2) %>%
  summarize(mean=mean(value),
            sd = sd(value))


```
```

```
res_cont
res_cont$export <- with(res_cont, sprintf("%.2f (%.2f)", mean, sd))

res_cont
```

```{r}
res_cont <- res_cont %>% select(-mean,-sd)
res_cont <- res_cont %>% pivot_wider(names_from = preterm_2, values_from = export)
```

```{r}
# Write this table to a comma separated .txt file:
write.table(res_cont, file = "desc_stat_cont.txt", sep = ",", quote = FALSE, row.names = F)
```

EDA for categorical variables

```{r}
p1 <- df %>%
  ggplot(mapping = aes(x = neonateSex)) +
  geom_bar(mapping = aes(fill = preterm_2),
           position = 'fill') +
  scale_fill_brewer(palette = "Pastel1") +
  theme_bw() + xlab("Neonate Sex") +ylab("")+
  theme(legend.position = "None")


p2 <- df %>%
  ggplot(mapping = aes(x = seasonBirth)) +
  geom_bar(mapping = aes(fill = preterm_2),
           position = 'fill') +
  scale_fill_brewer(palette = "Pastel1") +
  theme_bw() + xlab("Season of birth")+ylab("")

p3 <- df %>%
  ggplot(mapping = aes(x = momResCounty)) +
  geom_bar(mapping = aes(fill = preterm_2),
           position = 'fill') +
  scale_fill_brewer(palette = "Pastel1") +
  theme_bw() + xlab("Mother's residential county")+
  scale_x_discrete(labels=c("Allegheny",
                            "Beaver",
                            "Butler",
                            "Washing-\nton",
                            "Westmore-\nland",
```

```
                "Other"))+ylab(""))+
  theme(legend.position = "None")

p4 <- df %>%
  ggplot(mapping = aes(x = momEdu)) +
  geom_bar(mapping = aes(fill = preterm_2),
           position = 'fill') +
  scale_fill_brewer(palette = "Pastel1") +
  theme_bw() + xlab("Mother's education") +
  scale_x_discrete(labels=c("Associate\nDegree",
                "Bachelor's\nDegree",
                "Graduate/\nProfessional\nDegree",
                "HS/GED/\nSome college",
                "Less than HS"))+ylab("")

p5 <- df %>%
  ggplot(mapping = aes(x = momAge)) +
  geom_bar(mapping = aes(fill = preterm_2),
           position = 'fill') +
  scale_fill_brewer(palette = "Pastel1") +
  theme_bw() + xlab("Mother's age")+ylab("")+
  theme(legend.position = "None")

p6 <- df %>%
  ggplot(mapping = aes(x = momRace)) +
  geom_bar(mapping = aes(fill = preterm_2),
           position = 'fill') +
  scale_fill_brewer(palette = "Pastel1") +
  theme_bw() + xlab("Mother's race")+
  scale_x_discrete(labels=c("Black/\nAfrican American",
                "White",
                "Other"))+ylab("")

p7 <- df %>%
  ggplot(mapping = aes(x = momWIC)) +
  geom_bar(mapping = aes(fill = preterm_2),
           position = 'fill') +
  scale_fill_brewer(palette = "Pastel1") +
  theme_bw() + xlab("Mother's receipt of WIC services")+ylab("")+
  theme(legend.position = "None")

p8 <- df %>%
  ggplot(mapping = aes(x = momGestDiabetes)) +
  geom_bar(mapping = aes(fill = preterm_2),
           position = 'fill') +
  scale_fill_brewer(palette = "Pastel1") +
```

```
  theme_bw() + xlab("Diagnosis of gestational diabetes")+ylab("")

p9 <- df %>%
  ggplot(mapping = aes(x = smkPrior)) +
  geom_bar(mapping = aes(fill = preterm_2),
           position = 'fill') +
  scale_fill_brewer(palette = "Pastel1") +
  theme_bw() + xlab("Smoking prior to pregnancy")+ylab("")+
  theme(legend.position = "None")

p10 <- df %>%
  ggplot(mapping = aes(x = momPreviousLiveBirths)) +
  geom_bar(mapping = aes(fill = preterm_2),
           position = 'fill') +
  scale_fill_brewer(palette = "Pastel1") +
  theme_bw() + xlab("Number of previous live births")+ylab("")

p11 <- df %>%
  ggplot(mapping = aes(x = NOPV)) +
  geom_bar(mapping = aes(fill = preterm_2),
           position = 'fill') +
  scale_fill_brewer(palette = "Pastel1") +
  theme_bw() + xlab("Number of prenatal visits")+ylab("")
```

```{r}
plist <- list(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11)
```

```{r, fig.height=20}
library(gridExtra)

# display plot
g <- grid.arrange(grobs = plist, ncol = 2)
```

```{r}
# save plot
ggsave(file="birth_barplots.png", g, width = 275, height = 297, units = "mm")
```

EDA for continous variables

```{r}
```

```
q1 <- birth_data_w_co %>% ggplot(aes(x=mean_co)) + geom_freqpoly(mapping = aes(color = preterm_2, y=stat(density)))+ scale_color_brewer(palette = "Pastel1", direction=-1) + theme_bw() +
  theme(legend.position = "None")

q2 <- birth_data_w_no2 %>% ggplot(aes(x=mean_no2)) + geom_freqpoly(mapping = aes(color = preterm_2, y=stat(density)))+ scale_color_brewer(palette = "Pastel1", direction=-1)+ theme_bw()

q3 <- birth_data_w_o3 %>% ggplot(aes(x=mean_o3)) + geom_freqpoly(mapping = aes(color = preterm_2, y=stat(density)))+ scale_color_brewer(palette = "Pastel1", direction=-1)+ theme_bw() +
  theme(legend.position = "None")

q4 <- birth_data_w_pm2.5 %>% ggplot(aes(x=mean_pm2.5)) + geom_freqpoly(mapping = aes(color = preterm_2, y=stat(density)))+ scale_color_brewer(palette = "Pastel1", direction=-1)+ theme_bw()

q5 <- birth_data_w_so2 %>% ggplot(aes(x=mean_so2)) + geom_freqpoly(mapping = aes(color = preterm_2, y=stat(density)))+ scale_color_brewer(palette = "Pastel1", direction=-1)+ theme_bw()

```

```{r}
qlist <- list(q1,q2,q3,q4,q5)
```

```{r, fig.height=20}
library(gridExtra)

# display plot
h <- grid.arrange(grobs = qlist, ncol = 2)
```

```{r}
# save plot
ggsave(file="epa_freqpoly.png", h, width = 275, height = 297, units = "mm")
```
Empirical Bayes exploration was assisted by Dr. Joseph Yurko.
---
title: "Empirical Bayes exploration"
output: html_document

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

```

```{r}
library(tidyverse)
```

```{r}
df <- read_rds('df.rds')

dim(df)

glimpse(df)
```

```{r}
# all categorical variables are stored as cat_names
cat_names <- df %>%
  select(-preterm_2) %>%
  purrr::keep(is.factor) %>%
  names()

cat_group_summary <- df %>%
  group_by(across(all_of(cat_names))) %>%
  summarise(N = n(),
            m = sum(preterm_2 == 'yes'),
            mle = mean(preterm_2 == 'yes'),
            .groups = 'drop')

cat_group_summary %>% dim()
```

```{r}
# distribution for the number of births per group
cat_group_summary %>%
  ggplot(mapping = aes(x = N)) +
  geom_histogram(binwidth = 10) +
  theme_bw()
```

```{r}
# distribution of the preterm birth frequency (MLE)
cat_group_summary %>%
  ggplot(mapping = aes(x = mle)) +
  geom_histogram(binwidth = 0.02) +
  theme_bw()
```
```

```{r}
# how many groups have at least 50 births
cat_group_summary %>%
  filter(N > 49) %>%
  nrow()

# how many have only one
cat_group_summary %>%
  filter(N == 1) %>%
  nrow()
```

```{r}
# distribution of the preterm MLE for all groups that have at least 50 births
cat_group_summary %>%
  ggplot(mapping = aes(x = mle)) +
  geom_histogram(binwidth = 0.02) +
  facet_wrap(~ N > 49, labeller = 'label_both', scales = 'free_y') +
  theme_bw()
```

```{r}
# use empirical Bayes to craft a meaningful prior on the event probability

# use groups with at least 50 births
keep_groups <- cat_group_summary %>%
  filter(N > 49)

# need the more flexible beta-binomial distribution compared to the beta
# in order to perform empirical bayes
my_dbetabinomial_log <- function(m, N, a, b)
{
  lchoose(N, m) + lbeta(m + a, N - m + b) - lbeta(a, b)
}
```

```{r}
# define the log-likelihood function for the log-transformed a and b
my_loglik_betabinom <- function(unknowns, my_info)
{
  log_a <- unknowns[1]
  log_b <- unknowns[2]

  a <- exp(log_a)
  b <- exp(log_b)
```

```
  log_lik <- sum(purrr::map2_dbl(my_info$m, my_info$N,
                                 my_dbetabinomial_log,
                                 a = a,
                                 b = b))

  log_lik + log_a + log_b
}
```

```{r}
# setup the list of required information
info_use <- list(
  m = keep_groups$m,
  N = keep_groups$N
)

# run optim to estimate a and b
res_optim <- optim(c(0, 0),
                   my_loglik_betabinom,
                   gr = NULL,
                   info_use,
                   hessian = TRUE,
                   method = 'BFGS',
                   control = list(fnscale = -1, maxit = 1001))

res_optim

# check from another guess
res_check1 <- optim(c(1, 1),
                    my_loglik_betabinom,
                    gr = NULL,
                    info_use,
                    hessian = TRUE,
                    method = 'BFGS',
                    control = list(fnscale = -1, maxit = 1001))

res_check2 <- optim(c(0.5, -0.5),
                    my_loglik_betabinom,
                    gr = NULL,
                    info_use,
                    hessian = TRUE,
                    method = 'BFGS',
                    control = list(fnscale = -1, maxit = 1001))
```

```

```{r}
# compare
res_optim$par

res_check1$par

res_check2$par

## they are close enough
```

```{r}
empbayes_ab <- exp(res_optim$par)

empbayes_ab
```
```{r}
# a priori number of trials
sum(empbayes_ab)
```

```{r}
# plot the informative prior density
keep_groups %>%
  ggplot(mapping = aes(x = mle)) +
  geom_histogram(binwidth = 0.02,
                 mapping = aes(y = stat(density))) +
  stat_function(fun = dbeta,
                args = list(shape1 = empbayes_ab[1], shape2 = empbayes_ab[2]),
                xlim = c(0, 1),
                color = 'red', size = 1.2) +
  coord_cartesian(xlim = c(0, 1)) +
  theme_bw()

# this prior might not truly be suitable...the actual distribution of
# the mle's might be a mixture of betas!!!!
# there almost seems to be a second mode near 0.25?!?!?!

keep_groups %>%
  filter(between(mle, 0.2, 0.3)) %>%
  select(N, m, mle)
```
```

```{r}
# keep with this prior for now and see what happens

# the posterior shape parameters assuming a binomial likelihood and beta prior

cat_groups_post_2 <- cat_group_summary %>%
  mutate(a_new = m + empbayes_ab[1],
         b_new = N - m + empbayes_ab[2]) %>%
  mutate(post_avg = a_new / (a_new + b_new),
         post_q05 = qbeta(0.05, a_new, b_new),
         post_q95 = qbeta(0.95, a_new, b_new),
         post_prob_grt_0.1 = 1 - pbeta(0.1, a_new, b_new))

# plot the distribution of the posterior mean across all groups
cat_groups_post_2 %>%
  ggplot(mapping = aes(x = post_avg)) +
  geom_histogram(binwidth = 0.01) +
  theme_bw()

# plot the posterior summaries across all groups
cat_groups_post_2 %>%
  tibble::rowid_to_column("j") %>%
  mutate(j = as.factor(j)) %>%
  mutate(j = forcats::fct_reorder(j, post_avg, 'min')) %>%
  mutate(avg_group = cut(post_avg,
                         breaks = unique(quantile(post_avg)),
                         include.lowest = TRUE)) %>%
  ggplot(mapping = aes(x = j)) +
  geom_linerange(mapping = aes(group = j,
                               ymin = post_q05,
                               ymax = post_q95)) +
  geom_point(mapping = aes(group = j,
                           y = post_avg)) +
  facet_wrap(~avg_group, scales = 'free') +
  labs(x = '', y = 'event probability') +
  theme_bw() +
  theme(axis.text.x = element_blank(),
        strip.text = element_blank(),
        strip.background = element_blank(),
        panel.grid.major.x = element_blank())

```

This prior might be too strong due to that "upper" mode for the mle, so it's basically saying there could be two groups! A group with a very low event probability and another group with higher event probability.

```{r}
# but using this result, show the distribution of the posterior mean
# based on the separate categorical variables within the group

b1 <- cat_groups_post_2 %>%
  ggplot(mapping = aes(x = neonateSex, y = post_avg)) +
  geom_violin(fill = 'grey') +
  geom_boxplot(fill = NA) +
  theme_bw() + xlab("Neonate sex")

b2 <- cat_groups_post_2 %>%
  ggplot(mapping = aes(x = seasonBirth, y = post_avg)) +
  geom_violin(fill = 'grey') +
  geom_boxplot(fill = NA) +
  theme_bw() + xlab("Season of birth") +ylab("")

b3 <- cat_groups_post_2 %>%
  ggplot(mapping = aes(x = momResCounty, y = post_avg)) +
  geom_violin(fill = 'grey') +
  geom_boxplot(fill = NA) +
  theme_bw() + xlab("Mother's residential county") +
  scale_x_discrete(labels=c("Allegheny",
                "Beaver",
                "Butler",
                "Washing-\nton",
                "Westmore-\nland",
                "Other"))

b4 <- cat_groups_post_2 %>%
  ggplot(mapping = aes(x = momEdu, y = post_avg)) +
  geom_violin(fill = 'grey') +
  geom_boxplot(fill = NA) +
  theme_bw() + xlab("Mother's education") +
  scale_x_discrete(labels=c("Associate\nDegree",
                "Bachelor's\nDegree",
                "Graduate/\nProfessional\nDegree",
                "HS/GED/\nSome\ncollege",
                "Less\nthan HS"))+ylab("")

b5 <- cat_groups_post_2 %>%
  ggplot(mapping = aes(x = momAge, y = post_avg)) +
  geom_violin(fill = 'grey') +
```

```
  geom_boxplot(fill = NA) +
  theme_bw() + xlab("Mother's age")

b6 <- cat_groups_post_2 %>%
  ggplot(mapping = aes(x = momRace, y = post_avg)) +
  geom_violin(fill = 'grey') +
  geom_boxplot(fill = NA) +
  theme_bw() + xlab("Mother's race")+
  scale_x_discrete(labels=c("Black/\nAfrican American",
                "White",
                "Other"))+ylab("")

b7 <- cat_groups_post_2 %>%
  ggplot(mapping = aes(x = momWIC, y = post_avg)) +
  geom_violin(fill = 'grey') +
  geom_boxplot(fill = NA) +
  theme_bw() + xlab("Receipt of WIC services")

b8 <- cat_groups_post_2 %>%
  ggplot(mapping = aes(x = momGestDiabetes, y = post_avg)) +
  geom_violin(fill = 'grey') +
  geom_boxplot(fill = NA) +
  theme_bw() + xlab("Gestational diabetes")+ylab("")

b9 <- cat_groups_post_2 %>%
  ggplot(mapping = aes(x = smkPrior, y = post_avg)) +
  geom_violin(fill = 'grey') +
  geom_boxplot(fill = NA) +
  theme_bw() + xlab("Smoking prior to pregnancy")

b10 <- cat_groups_post_2 %>%
  ggplot(mapping = aes(x = momPreviousLiveBirths, y = post_avg)) +
  geom_violin(fill = 'grey') +
  geom_boxplot(fill = NA) +
  theme_bw() + xlab("Number of previous live births")+ylab("")

b11 <- cat_groups_post_2 %>%
  ggplot(mapping = aes(x = NOPV, y = post_avg)) +
  geom_violin(fill = 'grey') +
  geom_boxplot(fill = NA) +
  theme_bw() + xlab("Number of prenatal visits")
```

```{r}
blist <- list(b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11)
```

```{r, fig.height=20}
library(gridExtra)

# display plot
b <- grid.arrange(grobs = blist, ncol = 2)
```

```{r}
# save plot
ggsave(file="bayes_results.png", b, width = 275, height = 297, units = "mm")
```

```{r}
cat_groups_post_2 <- cat_groups_post_2 %>%
  mutate(lbirth_lumped = forcats::fct_lump_prop(LBIRTH, 0.05))

cat_groups_post_2 %>% count(mom_edu_collapsed)
cat_groups_post_2 %>%
  ggplot(mapping = aes(x = lbirth_lumped, y = post_avg)) +
  geom_violin(fill = 'grey') +
  geom_boxplot(fill = NA) +
  theme_bw() + xlab("Number of previous live births")
```

```{r}
# what's NOPV_c? highest posterior average for the event probability
# is observed when NOPV_c = 0-10
cat_groups_post_2 %>%
  select(NOPV_c, N, m, mle, post_avg, post_prob_grt_0.1) %>%
  arrange(desc(post_avg))

cat_groups_post_2 %>%
  count(NOPV_c)

cat_groups_post_2 %>%
  filter(post_avg > 0.12) %>%
  count(NOPV_c)
```

```{r}
# the groups with the highest posterior average
cat_groups_post_2 %>%
  arrange(desc(post_avg))
```

````
title: "Model Building: Downsampling"
output: html_document
---

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

```{r}
# required libraries
library(tidyverse)
```

```{r}
# read in dataset
df <- read_rds('df.rds')
```

221060 total observations

```{r}
b_data_caret <- df

#convert output to numeric
b_data_caret <- b_data_caret %>% mutate(preterm_2 = case_when(preterm_2 == 'yes'~ 1,
                                        preterm_2 == 'no'~ 0))

dmy <- dummyVars("~.", data=b_data_caret)
b_data_caret <- data.frame(predict(dmy, newdata = b_data_caret))

b_data_caret <-  b_data_caret %>% mutate(preterm_2 = case_when(preterm_2 == 1~'yes',
                                        preterm_2 == 0~'no'))

b_data_caret$preterm_2 <- as.factor(b_data_caret$preterm_2)
b_data_caret$preterm_2 <- relevel(b_data_caret$preterm_2, "yes")
```

```{r}
b_data_caret %>% count(preterm_2) %>% mutate(prop=n/sum(n))
```

```{r}
# install rsample
library(rsample)

set.seed(4321)
````
````

```
# choosing 70% of the data to be the training data
data_split_caret <- initial_split(b_data_caret, prop = .70)

# extracting training data and test data as two separate dataframes
data_train_caret <- training(data_split_caret)
data_test_caret  <- testing(data_split_caret)
```

```{r}
# training set proportions by preterm_2
data_train_caret %>% count(preterm_2) %>% mutate(prop=n/sum(n))

# testing set proportions by preterm_2
data_test_caret %>% count(preterm_2) %>% mutate(prop=n/sum(n))
```

```{r}
# use ROSE package to downsample
library(ROSE)
data_train_caret_bal <- ovun.sample(preterm_2~., data=data_train_caret, method="under", N=
22008)$data
table(data_train_caret_bal$preterm_2)
```

```{r}
ctrl <- trainControl(method = "repeatedcv", number = 5, repeats=3,
                     summaryFunction = twoClassSummary,
                     classProbs = TRUE,
                     savePredictions = TRUE)

```

ElaticNet

```{r}
enet_grid <- expand.grid(alpha = c(0.1, 0.2, 0.3, 0.4),
                         lambda = exp(seq(-6, 1, length.out = 21)))

set.seed(4321)
fit_glmnet_down <- train(preterm_2 ~ ., data = data_train_caret_bal,
                     method = "glmnet",
                     metric = "ROC",
                     preProcess = c('center','scale'),
                     tuneGrid = enet_grid,
                     trControl = ctrl)
fit_glmnet_down %>% readr::write_rds("fit_glmnet_down.rds")
```

```

```{r}
fit_glmnet_down
```

```{r}
plot(fit_glmnet_down, xTrans = log)
```

```{r}
fit_glmnet_down$bestTune
```

```{r}
confusionMatrix.train(fit_glmnet_down,positive = "yes")
```

Assessing model performance on the test set

```{r}
data_test_y <- data_test_caret$preterm_2
data_test_caret <- data_test_caret %>% select(-preterm_2)
glmnet_test <- predict(fit_glmnet_down, data_test_caret, type="prob")
glmnet_test_pred <- as.factor(ifelse(glmnet_test$yes > 0.5, "yes","no"))

glmnet_test_perf <- confusionMatrix(glmnet_test_pred, data_test_y, positive = "yes")
```

```{r}
roc.curve(data_test_y, glmnet_test[,2])
```

SVM Radial

```{r}
set.seed(4321)
fit_svm_down <- train(preterm_2 ~ ., data = data_train_caret_bal,
                      method = "svmRadial",
                      metric = "ROC",
                      preProcess = c('center','scale'),
                      trControl = ctrl)


fit_svm_down %>% readr::write_rds("fit_svm_down.rds")
```
```

```{r}
fit_svm_down$bestTune
```

```{r}
plot(fit_svm_down)
```

```{r}
confusionMatrix.train(fit_svm_down)
```

Assessing model performance on the test set

```{r}
svm_test <- predict(fit_svm_down, data_test_caret, type="prob")
svm_test_pred <- as.factor(ifelse(svm_test$yes > 0.5, "yes","no"))

svm_test_perf <- confusionMatrix(svm_test_pred, data_test_y, positive = "yes")
```

```{r}
roc.curve(data_test_y, svm_test[,2])
```

Random Forest

```{r}
set.seed(4321)
fit_rf_down <- train(preterm_2 ~ ., data = data_train_caret_bal,
                     method = "rf",
                     metric = "ROC",
                     preProcess = c('center','scale'),
                     trControl = ctrl,
                     importance = TRUE)


fit_rf_down %>% readr::write_rds("fit_rf_down.rds")
```

```{r}
fit_rf_down
```

```{r}
```

```
fit_rf_down$bestTune
```

```{r}
confusionMatrix.train(fit_rf_down)
```

Assessing model performance on the test set

```{r}
rf_test <- predict(fit_rf_down, data_test_caret, type="prob")
rf_test_pred <- as.factor(ifelse(rf_test$yes > 0.5, "yes","no"))

rf_test_perf <- confusionMatrix(rf_test_pred, data_test_y, positive = "yes")
```

```{r}
roc.curve(data_test_y, rf_test[,2])
```

```{r}
#read in models

fit_glmnet_down <- read_rds("fit_glmnet_down.rds")
fit_svm_down <- read_rds("fit_svm_down.rds")
fit_rf_down <- read_rds("fit_rf_down.rds")
```

On the training set, comparing models:

```{r}
model_cv_res <- resamples(list(GLMNET = fit_glmnet_down,
                           SVM = fit_svm_down,
                           RF = fit_rf_down))
```

```{r}
cv_pred_results <- fit_glmnet_down$pred %>% tbl_df() %>%
          filter(alpha %in% fit_glmnet_down$bestTune$alpha,
               lambda %in% fit_glmnet_down$bestTune$lambda) %>%
          select(pred, obs, yes, no, rowIndex, Resample) %>%
          mutate(model_name = "GLMNET") %>%
  bind_rows(fit_svm_down$pred %>% tbl_df() %>%
          filter(sigma %in% fit_svm_down$bestTune$sigma,
               C %in% fit_svm_down$bestTune$C) %>%
```

```
            select(pred, obs, yes, no, rowIndex, Resample) %>%
            mutate(model_name = "SVM")) %>%
  bind_rows(fit_rf_down$pred %>% tbl_df() %>%
            filter(mtry == fit_rf_down$bestTune$mtry) %>%
            select(pred, obs, yes, no, rowIndex, Resample) %>%
            mutate(model_name = "RF"))
```

```{r}
library(plotROC)
```

```{r}
calC_auc(cv_pred_results)
```

```{r}
auc_train <- calc_auc(cv_pred_results %>%
  ggplot(mapping = aes(m = yes,
                       d = ifelse(obs == "yes",
                                  1,
                                  0),
                       color = model_name)) +
  geom_roc(cutoffs.at = 0.5) +
  coord_equal() +
  style_roc() +
  ggthemes::scale_color_colorblind())

cv_pred_results %>%
  ggplot(mapping = aes(m = yes,
                       d = ifelse(obs == "yes",
                                  1,
                                  0),
                       color = model_name)) +
  geom_roc(cutoffs.at = 0.5) +
  coord_equal() +
  style_roc() +
  ggthemes::scale_color_colorblind()

ggsave('auc_train.png')
```

```{r}
resamples_lf <- as.data.frame(model_cv_res, metric = "Sens") %>% tbl_df() %>%
  mutate(metric_name = "Sensitivity") %>%
  bind_rows(as.data.frame(model_cv_res, metric = "Spec") %>% tbl_df() %>%
```

```
            mutate(metric_name = "Specificity")) %>%
  tidyr::gather(key = "model_name", value = "metric_value",
                -Resample, -metric_name)
```

```{r}
resamples_lf%>%
group_by(metric_name,model_name)%>%
summarize(mean=mean(metric_value))
```

```{r}
resamples_lf %>%
  ggplot(mapping = aes(x = fct_reorder(model_name,metric_value), y = metric_value)) +
  stat_summary(fun.data = "mean_se",
               color = "red",
               fun.args = list(mult = 1)) +
  coord_flip() +
  facet_grid(. ~ metric_name, scales = "free_x") +
  theme_bw() +xlab("")

ggsave('sens_train.png')
```

Performance metrices for testing dataset

```{r}
roc_glmnet <- roc(data_test_y, glmnet_test[,2])
roc_svm <- roc(data_test_y, svm_test[,2])
roc_rf <- roc(data_test_y, rf_test[,2])
```

```{r}
ggroc(list(roc_glmnet, roc_svm, roc_rf), size=.8)+
  scale_color_manual(labels = c(paste0('GLMNET: ', round(roc_glmnet$auc,3)),
                                paste0('SVM: ', round(roc_svm$auc,3)),
                                paste0('RF: ', round(roc_rf$auc,3))),
                     values=c("#000000","#56B4E9","#E69F00")) +
  labs(color='') +
  theme_bw() +xlab("False Positive Fraction")+ylab("True Positive Fraction")

ggsave('auc_test.png')
```

```{r}
glmnetImp <- varImp(fit_glmnet_down, scale = TRUE)
```

```
plot(glmnetImp, top = 10)
```

```{r}
svmImp <- varImp(fit_svm_down, scale = TRUE)
plot(svmImp, top = 10)
```

```{r}
rfImp <- varImp(fit_rf_down, scale = TRUE)
plot(rfImp, top = 10)
```

Bibliography

Bekkar, B., Pacheco, S., Basu, R., & DeNicola, N. (2020). Association of air pollution and heat exposure with preterm birth, low birth weight, and stillbirth in the US. *JAMA Network Open*, *3*(6). https://doi.org/10.1001/jamanetworkopen.2020.8243

Blencowe, H., Cousens, S., Chou, D., Oestergaard, M., Say, L., Moller, A.-B., Kinney, M., & Lawn, J. (2013). Born too soon: The global epidemiology of 15 million preterm births. *Reproductive Health*, *10*(S1). https://doi.org/10.1186/1742-4755-10-s1-s2

Boehmke, B., & Greenwell, B. (2020). *Hands-on machine learning with R*. CRC Press.

Cantarutti, A., Franchi, M., Monzio Compagnoni, M., Merlino, L., & Corrao, G. (2017). Mother's education and the risk of several neonatal outcomes: An evidence from an Italian population-based study. *BMC Pregnancy and Childbirth*, *17*(1). https://doi.org/10.1186/s12884-017-1418-1

Goldenberg, R. L., Culhane, J. F., Iams, J. D., & Romero, R. (2008). Epidemiology and causes of preterm birth. *The Lancet*, *371*(9606), 75–84. https://doi.org/10.1016/s0140-6736(08)60074-4

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An introduction to statistical learning: With applications in R* (2nd ed.). Springer.

Kubat, M., Holte, R. C., & Matwin, S. (1998). Machine Learning for the Detection of Oil Spills in Satellite Radar Images. *Machine Learning*, *30*(2/3), 195–215. https://doi.org/10.1023/a:1007452223027

Leevy, J. L., Khoshgoftaar, T. M., Bauder, R. A., & Seliya, N. (2018). A survey on addressing high-class imbalance in Big Data. *Journal of Big Data*, *5*(1). https://doi.org/10.1186/s40537-018-0151-6

March of Dimes. (n.d.). Prematurity profile. Retrieved April 3, 2022, from https://www.marchofdimes.org/peristats/tools/prematurityprofile.aspx?reg=99

Martin, J. A., Hamilton, B. E., Osterman, M. J.K., Driscoll, A. K., & Drake, P. (2018). *Births: Final Data for 2016*. National Vital Statistics Reports. https://www.cdc.gov/nchs/data/nvsr/nvsr67/nvsr67_01.pdf

Oliveira, K. A., Araújo, E. M., Oliveira, K. A., Casotti, C. A., Silva, C., & Santos, D. (2018). Association between race/skin color and premature birth: a systematic review with meta-analysis. *Revista de saude publica, 52*, 26. https://doi.org/10.11606/S1518-8787.2018052000406

Sagiv, S. K., Mendola, P., Loomis, D., Herring, A. H., Neas, L. M., Savitz, D. A., & Poole, C. (2005). A time series analysis of air pollution and preterm birth in Pennsylvania, 1997–2001. *Environmental Health Perspectives, 113*(5), 602–606. https://doi.org/10.1289/ehp.7646

Sarica, A., Cerasa, A., & Quattrone, A. (2017). Random Forest algorithm for the classification of neuroimaging data in alzheimer's disease: A systematic review. *Frontiers in Aging Neuroscience, 9*. https://doi.org/10.3389/fnagi.2017.00329

Support Vector Machines. (n.d.). Retrieved March 30, 2022, from https://scikit-learn.org/stable/modules/svm.html

Vintzileos, A. M., Ananth, C. V., Smulian, J. C., Scorza, W. E., & Knuppel, R. A. (2002). The impact of prenatal care in the United States on preterm births in the presence and absence of antenatal high-risk conditions. *American Journal of Obstetrics and Gynecology*, *187*(5), 1254–1257. https://doi.org/10.1067/mob.2002.127140

Wang, R., Pan, W., Jin, L., Li, Y., Geng, Y., Gao, C., Chen, G., Wang, H., Ma, D., & Liao, S. (2019). Artificial Intelligence in reproductive medicine. *Reproduction*, *158*(4). https://doi.org/10.1530/rep-18-0523

World Health Organization. (n.d.). *Preterm birth*. World Health Organization. Retrieved April 3, 2022, from https://www.who.int/news-room/fact-sheets/detail/preterm-birth