CWRU Data Bootcamp: Final Project Presentation

Machine Learning and Healthcare

Presented by Team: Carol, Joan, Kashifa, Mohommed and Ratnam

Team Demo Day

THINGS TO COVER

- Who We Are
- What Our Project is About
- Why Infant Mortality is Important
- How Machine Learning Can Help
- Where Does the Microbiome Fit In
- Learn more about how we completed this project

FINAL PROJECT• 2019

Who We Are ...

OUR TEAM:

DATA SCIENCE STUDENTS

V. Ratnam Mantripragada, Joan E. Stone-Mays, Carol Kadish, Kashifa Ahmed and Mohammed N. Irshad

OUR GOALS FOR THIS PROJECT

We want to use data to explore and learn how data science can have a positive impact on healthcare outcomes for infants. Our focus is on the factors related to infant mortality and the microbiome.

Machine Learning: Maternal-Infant Healthcare

IDENTIFYING FACTORS, INCLUDING THE MICROBIOME TO IMPROVE INFANT MORTALITY

FINAL PROJECT • 2019

What is Infant Mortality?

THE DEATH OF A CHILD UNDER THE AGE OF 1. THE INFANT MORTALITY RATE (IMR) IS THE NUMBER OF INFANT DEATHS PER 1000 LIVE BIRTHS.

The IMR's vary greatly per location, maternal, labor and delivery factors as well as time and other variables.

What would we like to learn?

KEY QUESTIONS

What factors impact infant mortality

What factors can we use Machine Learning (ML) to predict

How can we use data and ML to improve infant mortality

CDC.gov

QUOTES ON THE IMPORTANCE OF INFANT MORTALITY

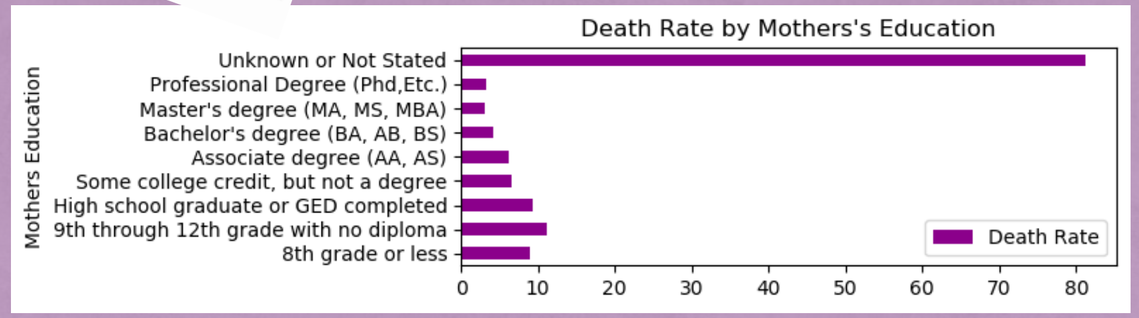
Infant mortality gives us key information about maternal and infant health, the infant mortality rate is an important marker of the overall health of a society.
The CDC (Centers for Disease Control) is committed to improving birth outcomes.

WHAT IS THE CURRENT INFANT MORTALITY RATE?

5.8 in the US
7.3 in Ohio
2018

INFANT MORTALITY RATES ARE A KEY FACTOR IN MEASURING THE HEALTH OF A SOCIETY

Three Groups of Testing Variables



DIFFERENCES IN DELIVERY FACTORS

Month of delivery, Birthplace, Delivery method, Fetal presentation

DIFFERENCES IN LABOR FACTORS

Anesthesia, Antibiotics used for Mom, Induction of labor, Steroid

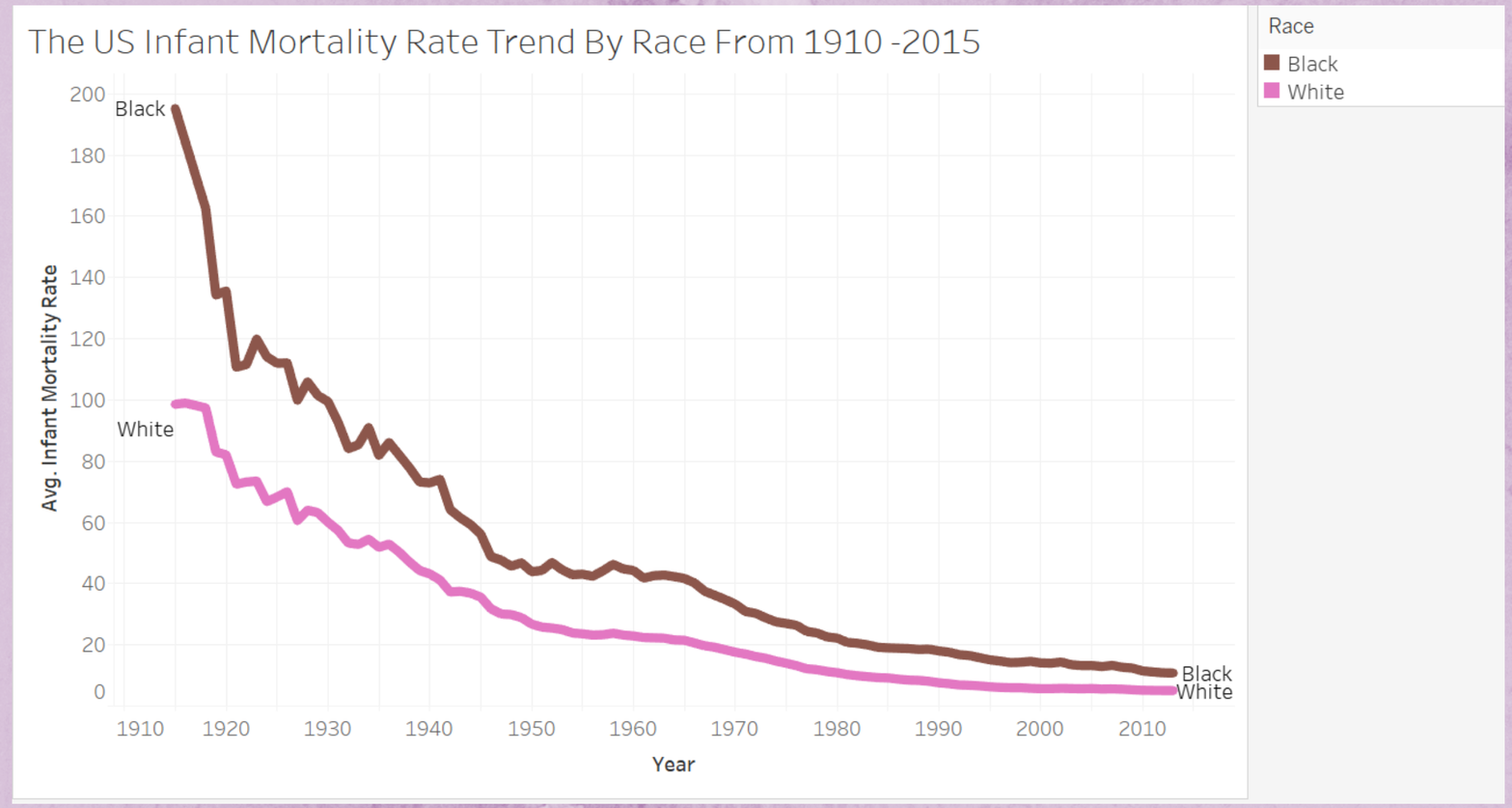
DIFFERENCES IN MATERNAL FACTORS

Mother's Education, Weight gain, pre-pregnancy diabetes / hypertension

Some factors were in EVERY set of data such as A mothers age ...

...average prenatal visits, prior births, Pre-pregnancy BMI ...

Babies weight in grams, gestational age in weeks and the five min APGAR score

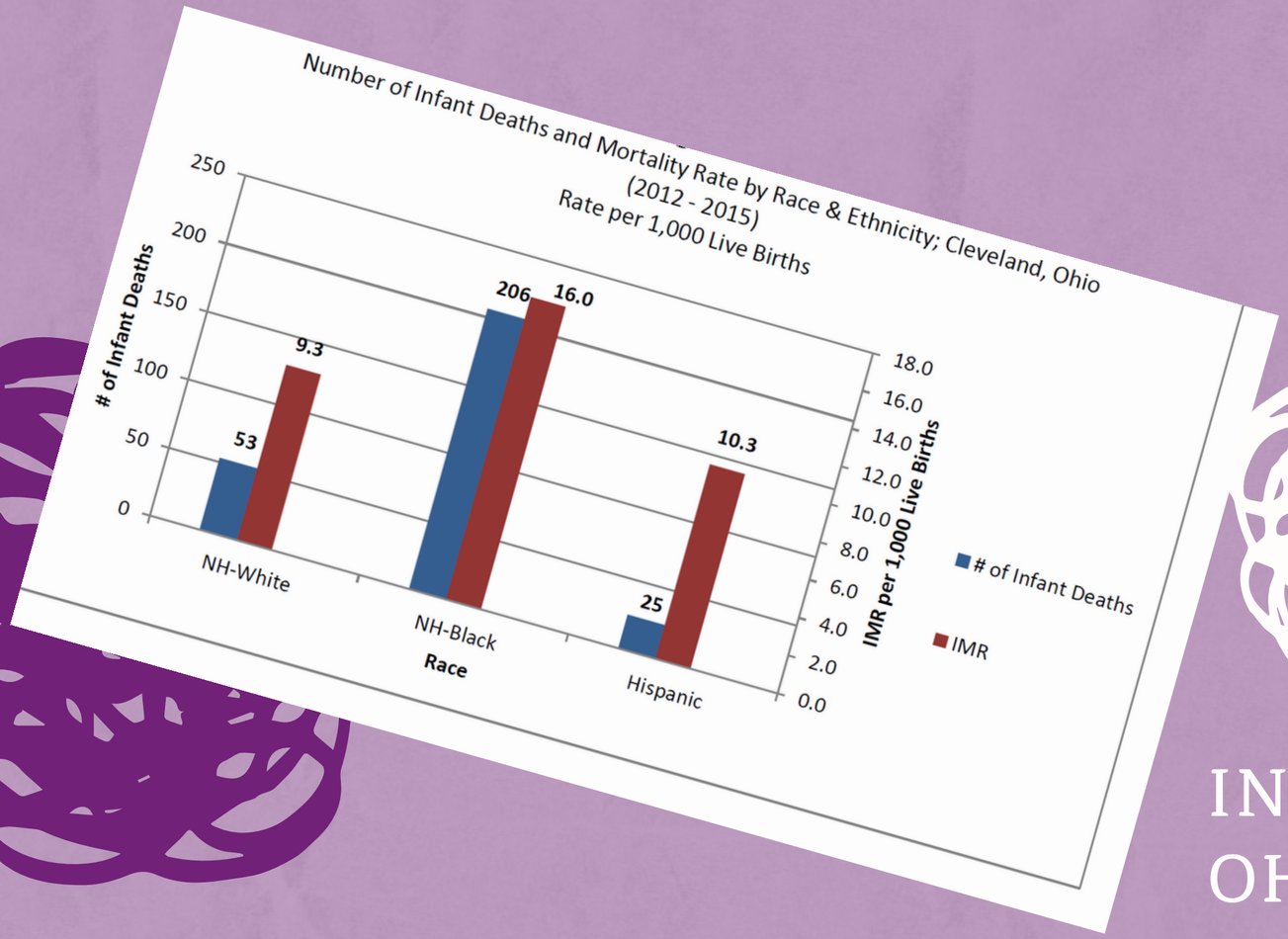


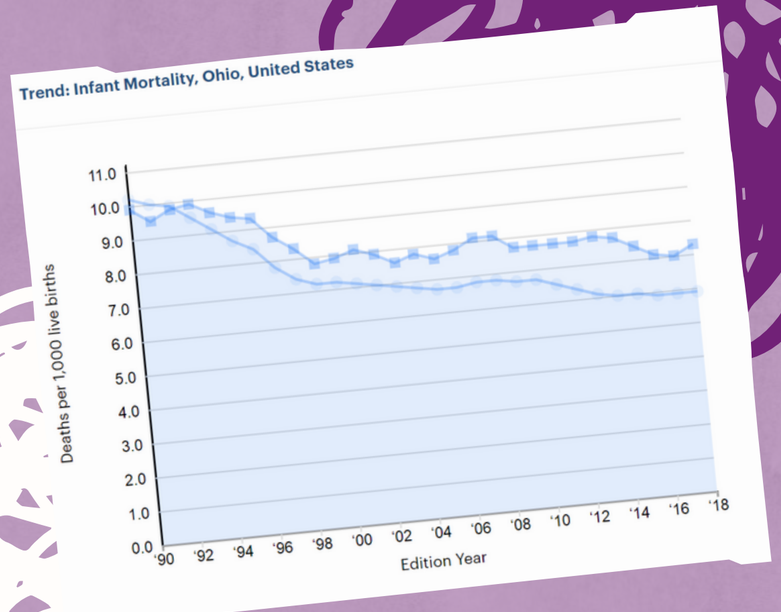
Historical Trends

RATES BY RACE

Line Graph of Infant mortality rates from 1910 - 2015, comparing race

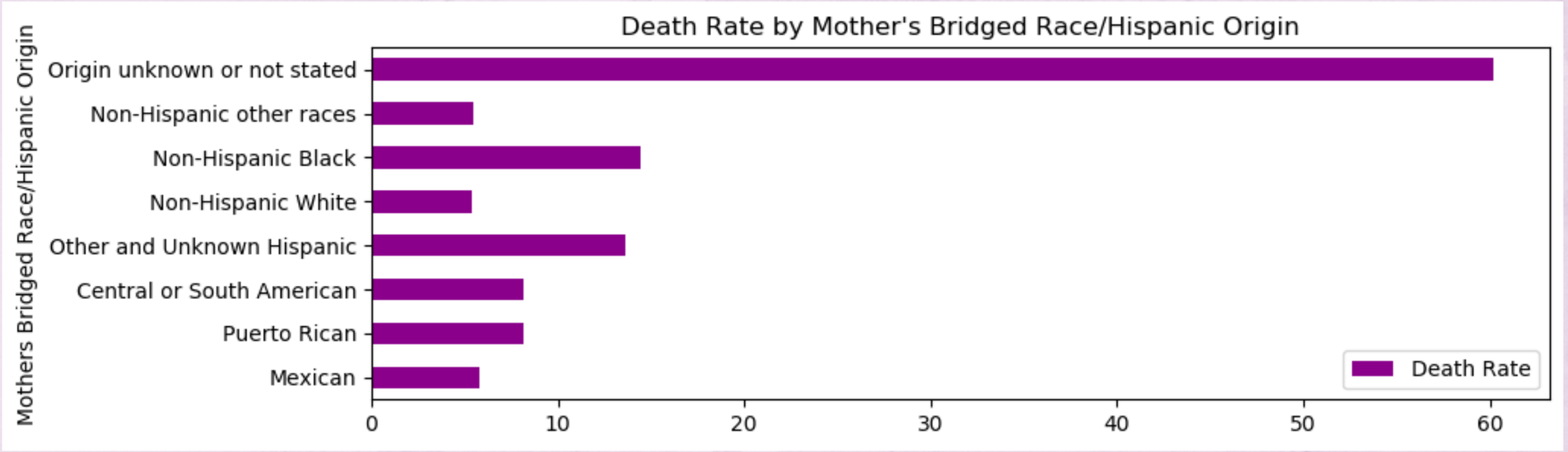
GAPS IN IMR RATES



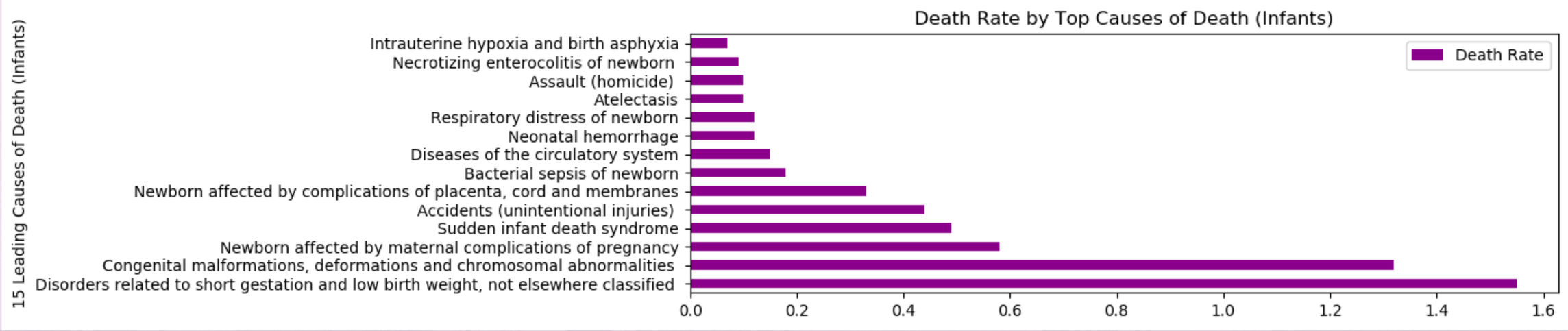


INFANT MORTALITY IN OHIO

Ohio trend-line is moving upwards this year. The gap between national trend and Ohio is growing.



Race and Ethnicity play a role in the Average IMR



The top 15 causes of infant mortality. The number one, Low Birth Weight, is linked to maternal health / behavior issues.

Where did we find our data?

GOVERNMENT AGENCIES (BOTH LOCAL AND NATIONAL). OUR MAIN SOURCE WAS THE CDC WONDER DATA.

Some of our research was reviewing published graphs, medical and science journals, to see if we could find their specific data sources cited.

What Are APGAR Scores?

APGAR scores are a health measurement taken at birth, five minutes and if needed, 10 mins after birth

The scale of 0-10 indicates if there is a need for medical intervention for the infant. 10 being the best possible score and the average between 7-9.

This was the best ANSWER to be found in the data we had available to us, to determine which factors impacted infant mortality.

APPEARANCE / SKIN COLOR

PULSE / HEART RATE

GRIMACE RESPONSE / REFLEXES

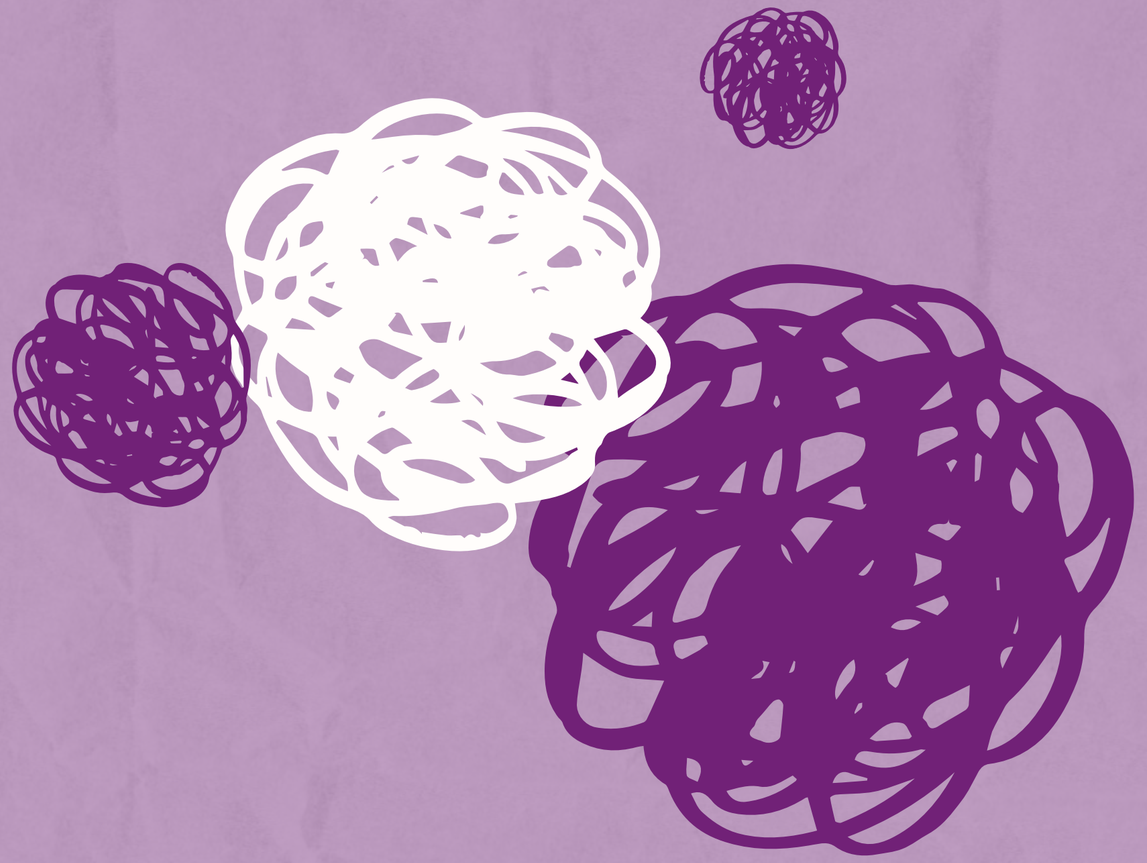
ACTIVITY / MUSCLE TONE

RESPIRATIONS / BREATHING RATE AND EFFORT

APGAR Scores

HOW ARE THEY CALCULATED? EACH OF THE LETTERS REPRESENTS A FEATURE THAT GETS SCORE POINTS, BASED ON WHAT THE MEDICAL PROFESSIONAL EVALUATES AND THE SUM OF THE POINTS CREATES THE SCORE.

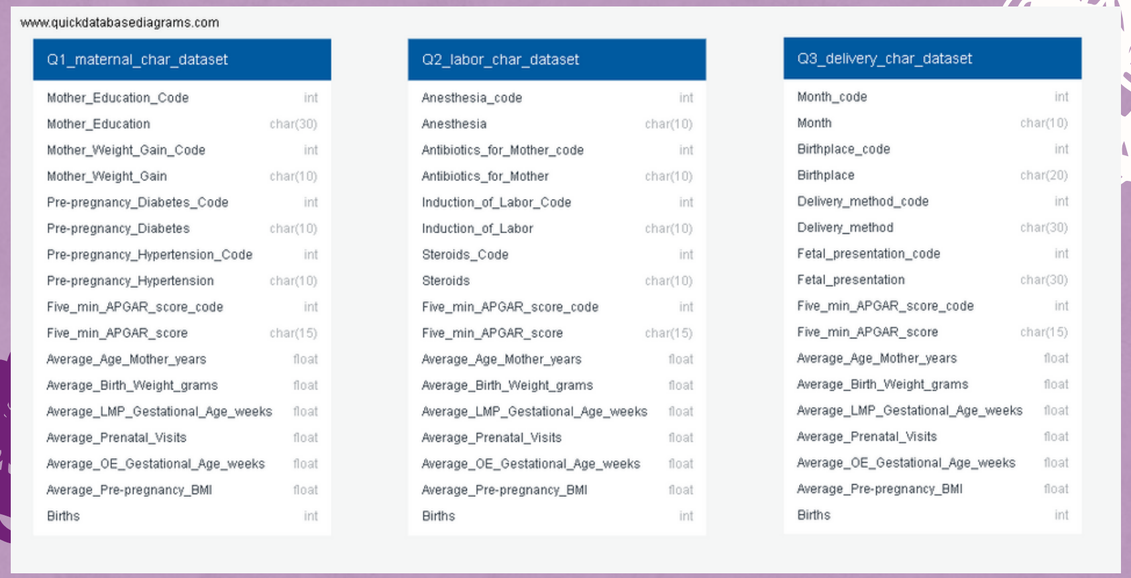
FINAL PROJECT • 2019



Machine Learning

OUR TESTS INCLUDED VARIOUS DECISION TREE MODELS SUCH AS, RANDOM FOREST AND XGBOOST

The best final accuracy score was 85.46%.



What kind of data did we get to model?

CATEGORICAL

distinct groups with no logical order, coded data

CONTINUOUS

numerical variables that have an infinite number of values, such as age

ISSUES

Our data had null values that didn't make sense, odd scoring values such as 99 for unknown that needed to be removed as well as needing to be restructured so that the algorithms could understand the answer variable.

Machine Learning

DEFINE A PROBLEM AND FIND DATA

Machine Learning needs specific kinds of data for specific models. Our problem needed clear definition before we could find data.

DATA IS DIRTY AND NEEDS LOTS OF CLEANING

Machine algorithms can be quite particular about what data it will respond to, therefore lots of cleaning and understanding of the data is needed.

SPLIT, TRAIN AND TEST YOUR MODEL

Once the "answer" has been removed, the data must be split into groups for training and then testing to see if the model has responded as expected.

REVIEW AND REVISE

If at first your model doesn't work, revise the hyper-parameters, review the model and continue to tweak and test until you have results.

RANDOM FOREST

We chose Random Forest for our model after trying other models, such as linear regression, as this is a learning process. Random Forest uses random decision trees to isolate knowledge with different applied variable arrays.

SUCCESSFUL MODELS

When we didn't get the accuracy results we needed, we went back and adjusted with get.dummies, which helps create dummy coded data. When this also failed we tried XGBoost, extreme gradient boost and at first only got 38% accuracy.
Then we created a binary for our answer binning the variable into two categories only <5 and >5 for the APGAR score.

Extreme Gradient Boosting

XGBOOST - WHAT IS IT?

Tianqi Chen - "refers to the engineering goal to push the limit of computations resources for boosted tree algorithms."

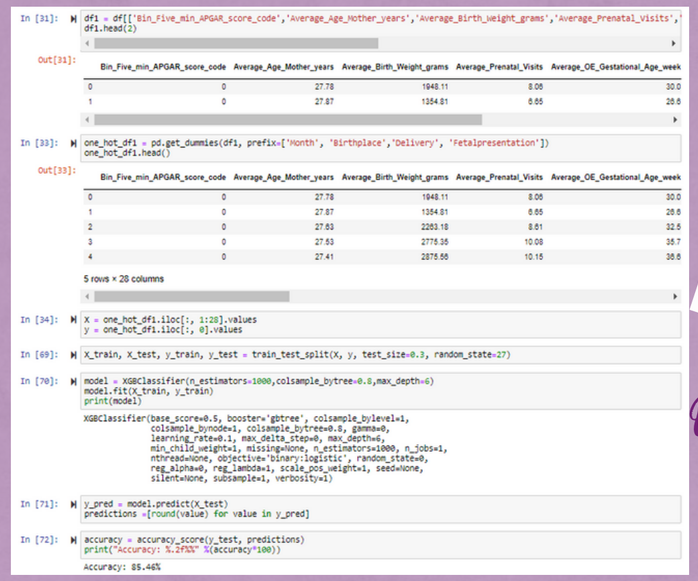
THE ALGORITHM FEATURES:

Sparse Aware with automatic handling of missing data values.
Block Structure to support parallelization of tree construction.
Continued training so you can boost an already fitted model on new data.

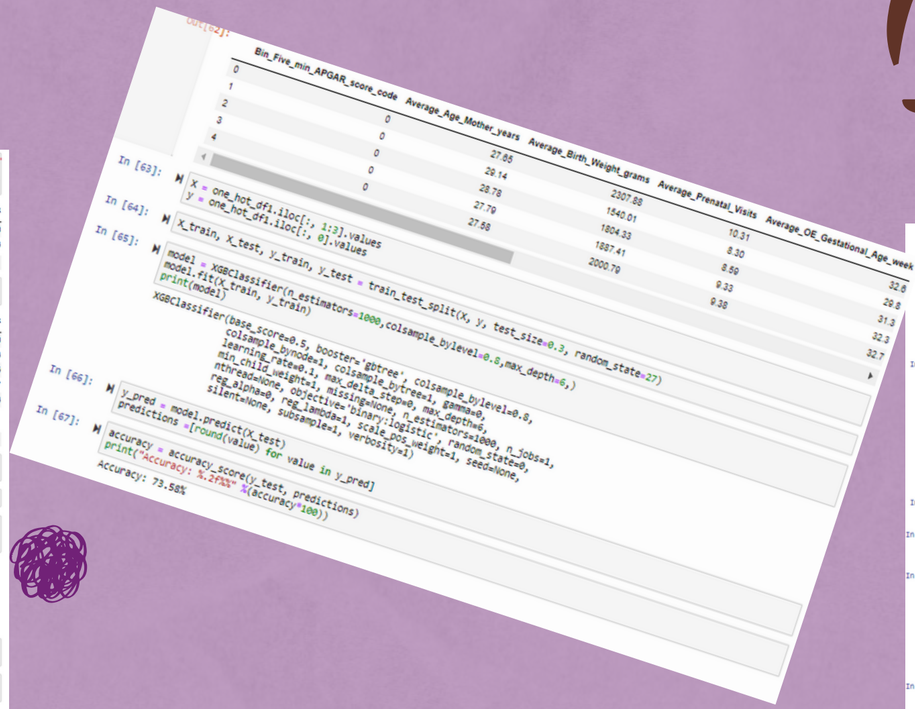
WHAT IS XGBOOST, BOOSTING?

Boosting is a technique where new models are added to correct the errors made by existing models. When no further improvements can be made, it stops adding models.

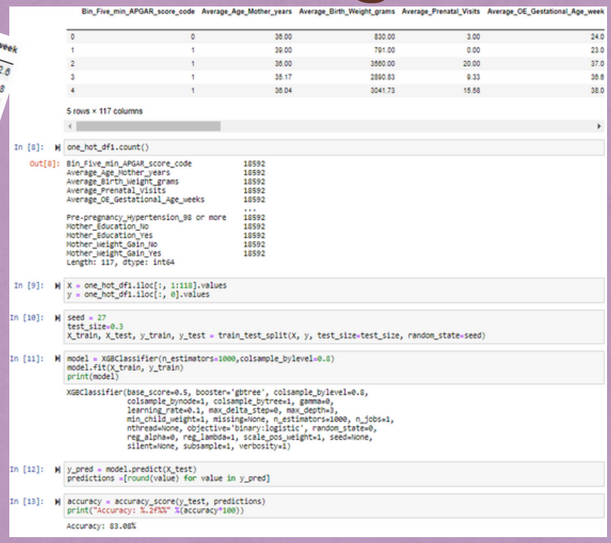
Three Groups of Testing Results



```
In [31]: df1 = df[['Bin_Five_min_APGAR_score_code','Average_Age_Mother_years','Average_Birth_Weight_grams','Average_Prenatal_Visits',
         df1.head(2)

Out[31]:
   Bin_Five_min_APGAR_score_code  Average_Age_Mother_years  Average_Birth_Weight_grams  Average_Prenatal_Visits  Average_OE_Gestational_Age_week
0                              0                     27.78                      1948.11                     8.06                             30.0
1                              0                     27.87                      1354.81                     6.65                             28.6

In [33]: one_hot_df1 = pd.get_dummies(df1, prefix=['Month', 'Birthplace','Delivery', 'Fetalpresentation'])
         one_hot_df1.head()

Out[33]:
   Bin_Five_min_APGAR_score_code  Average_Age_Mother_years  Average_Birth_Weight_grams  Average_Prenatal_Visits  Average_OE_Gestational_Age_week
0                              0                     27.78                      1948.11                     8.06                             30.0
1                              0                     27.87                      1354.81                     6.65                             28.6
2                              0                     27.63                      2283.18                     8.61                             32.6
3                              0                     27.63                      2776.35                    10.08                             35.7
4                              0                     27.41                      2875.56                    10.15                             36.6

5 rows × 28 columns

In [34]: X = one_hot_df1.iloc[:, 1:28].values
         y = one_hot_df1.iloc[:, 0].values

In [69]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=27)

In [70]: model = XGBClassifier(n_estimators=1000,colsample_bytree=0.8,max_depth=6)
         model.fit(X_train, y_train)
         print(model)

         XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
                       colsample_bynode=1, colsample_bytree=0.8, gamma=0,
                       learning_rate=0.1, max_delta_step=0, max_depth=6,
                       min_child_weight=1, missing=None, n_estimators=1000, n_jobs=1,
                       nthread=None, objective='binary:logistic', random_state=0,
                       reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
                       silent=None, subsample=1, verbosity=1)

In [71]: y_pred = model.predict(X_test)
         predictions = [round(value) for value in y_pred]

In [72]: accuracy = accuracy_score(y_test, predictions)
         print("Accuracy: %.2f%%" %(accuracy*100))

         Accuracy: 85.46%
```



```
Out[62]:
   Bin_Five_min_APGAR_score_code  Average_Age_Mother_years  Average_Birth_Weight_grams  Average_Prenatal_Visits  Average_OE_Gestational_Age_week
0                              0                     27.65                     2307.88                    10.31                             32.6
1                              0                     29.14                     1540.01                     8.30                             29.8
2                              0                     28.78                     1804.33                     8.59                             31.3
3                              0                     27.79                     1887.41                     9.33                             32.3
4                              0                     27.58                     2000.79                     9.38                             32.7

In [63]: X = one_hot_df1.iloc[:, 1:3].values
         y = one_hot_df1.iloc[:, 0].values

In [64]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=27)

In [65]: model = XGBClassifier(n_estimators=1000,colsample_bylevel=0.8,max_depth=6,)
         model.fit(X_train, y_train)
         print(model)

         XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=0.8,
                       colsample_bynode=1, colsample_bytree=1, gamma=0,
                       learning_rate=0.1, max_delta_step=0, max_depth=6,
                       min_child_weight=1, missing=None, n_estimators=1000, n_jobs=1,
                       nthread=None, objective='binary:logistic', random_state=0,
                       reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
                       silent=None, subsample=1, verbosity=1)

In [66]: y_pred = model.predict(X_test)
         predictions = [round(value) for value in y_pred]

In [67]: accuracy = accuracy_score(y_test, predictions)
         print("Accuracy: %.2f%%" %(accuracy*100))

         Accuracy: 73.58%
```



```
   Bin_Five_min_APGAR_score_code  Average_Age_Mother_years  Average_Birth_Weight_grams  Average_Prenatal_Visits  Average_OE_Gestational_Age_week
0                              0                     38.00                      830.00                     3.00                             24.0
1                              1                     39.00                      791.00                     0.00                             23.0
2                              1                     38.00                     3660.00                    20.00                             37.0
3                              1                     38.17                     2890.83                     9.33                             38.6
4                              1                     38.04                     3041.73                    15.58                             38.0

5 rows × 117 columns

In [8]: one_hot_df1.count()

Out[8]: Bin_Five_min_APGAR_score_code          18592
        Average_Age_Mother_years               18592
        Average_Birth_Weight_grams             18592
        Average_Prenatal_Visits                18592
        Average_OE_Gestational_Age_weeks       18592
                                               ...
        Pre-pregnancy_Hypertension_No or more  18592
        Mother_Education_No                    18592
        Mother_Education_Yes                   18592
        Mother_Weight_Gain_No                  18592
        Mother_Weight_Gain_Yes                 18592
        Length: 117, dtype: int64

In [9]: X = one_hot_df1.iloc[:, 1:118].values
        y = one_hot_df1.iloc[:, 0].values

In [10]: seed = 27
         test_size=0.3
         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=test_size, random_state=seed)

In [11]: model = XGBClassifier(n_estimators=1000,colsample_bylevel=0.8)
         model.fit(X_train, y_train)
         print(model)

         XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=0.8,
                       colsample_bynode=1, colsample_bytree=1, gamma=0,
                       learning_rate=0.1, max_delta_step=0, max_depth=3,
                       min_child_weight=1, missing=None, n_estimators=1000, n_jobs=1,
                       nthread=None, objective='binary:logistic', random_state=0,
                       reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
                       silent=None, subsample=1, verbosity=1)

In [12]: y_pred = model.predict(X_test)
         predictions = [round(value) for value in y_pred]

In [13]: accuracy = accuracy_score(y_test, predictions)
         print("Accuracy: %.2f%%" %(accuracy*100))

         Accuracy: 83.08%
```

DELIVERY FACTORS

Final Accuracy Score: 85.46%

Month of delivery, Birthplace, Delivery method, Fetal presentation

LABOR FACTORS

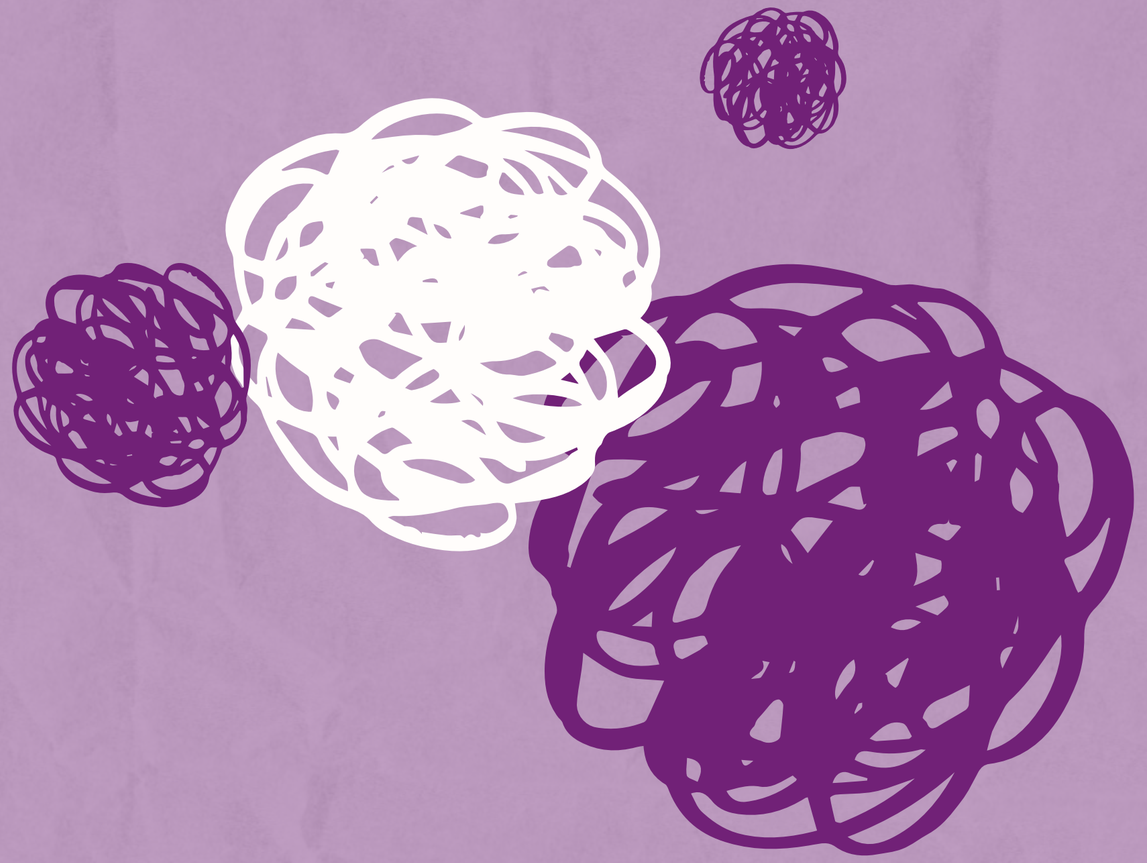
Final Accuracy Score: 73.58%

Anesthesia, Antibiotics used for Mom, Induction of labor, Steroid

MATERNAL FACTORS

Final Accuracy Score: 83.08%

Mother's Education, Weight gain, pre-pregnancy diabetes / hypertension



Microbiota / Microbiome

MICROBIOTA IS MADE UP OF TRILLIONS OF CELLS, BACTERIA, VIRUSES AND FUNGI THAT LIVE WITH AND INSIDE HUMANS.

They are vital to our health and impact our nutrition, immunity and effect the brain and our behavior.

20,000 - 25,000 protein coding genes

THE NUMBER OF PROTEIN CODING GENES FOUND BY THE HUMAN GENOME PROJECT.

8 million microbial genes

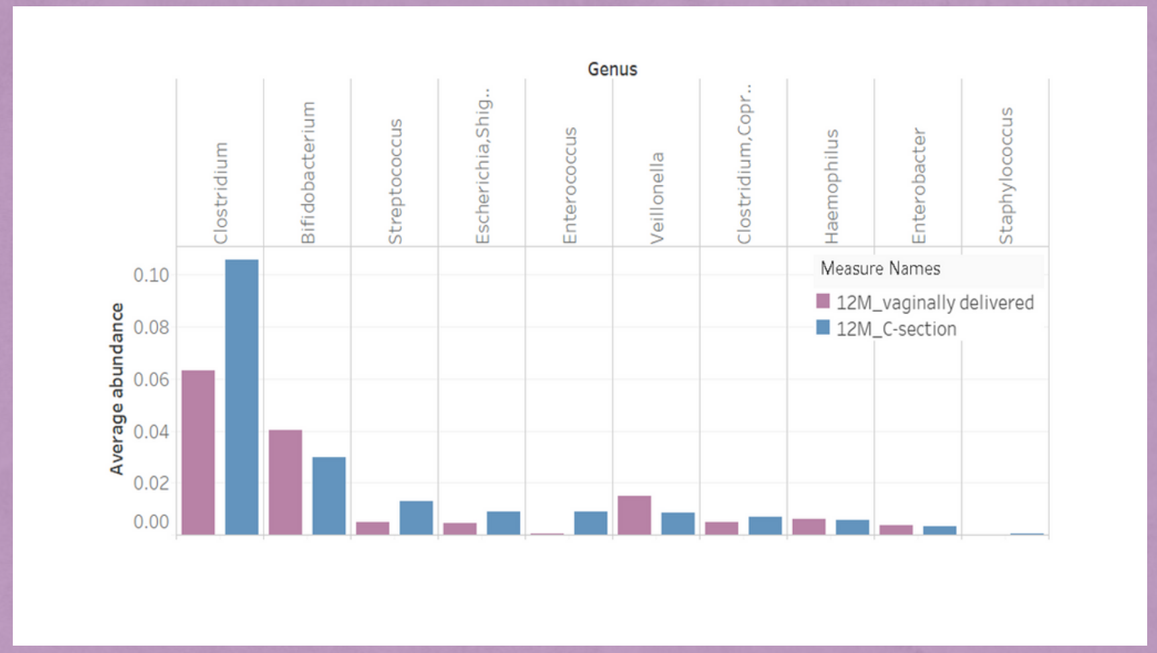
THE NUMBER OF UNIQUE GENES FOUND BY THE HUMAN MICROBIOME PROJECT

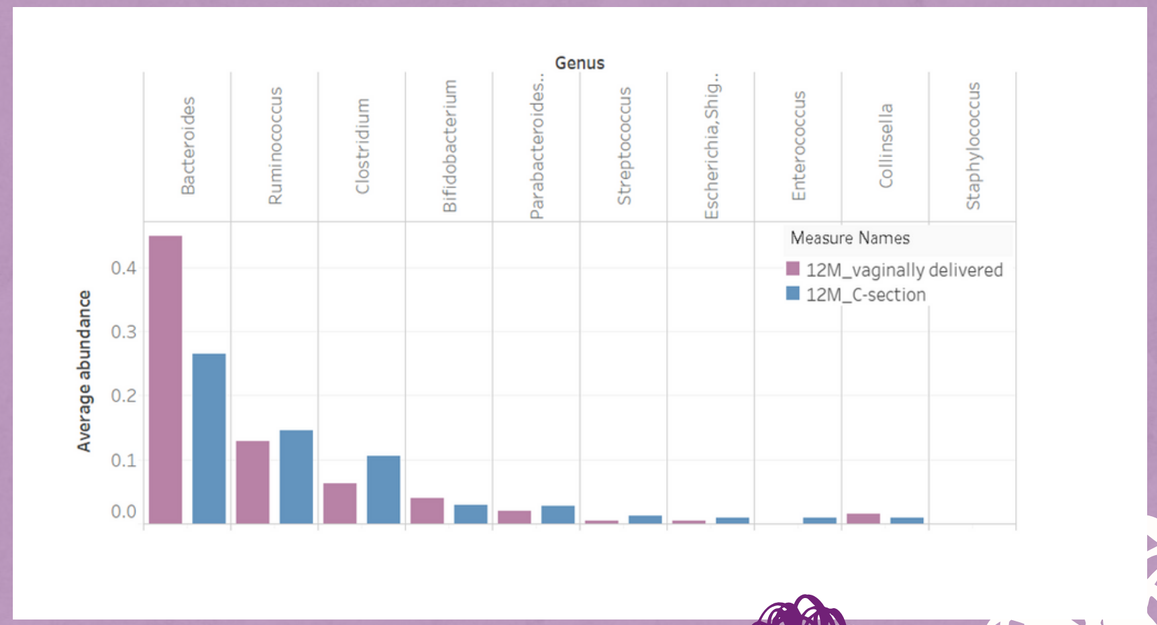
Microbiome Data

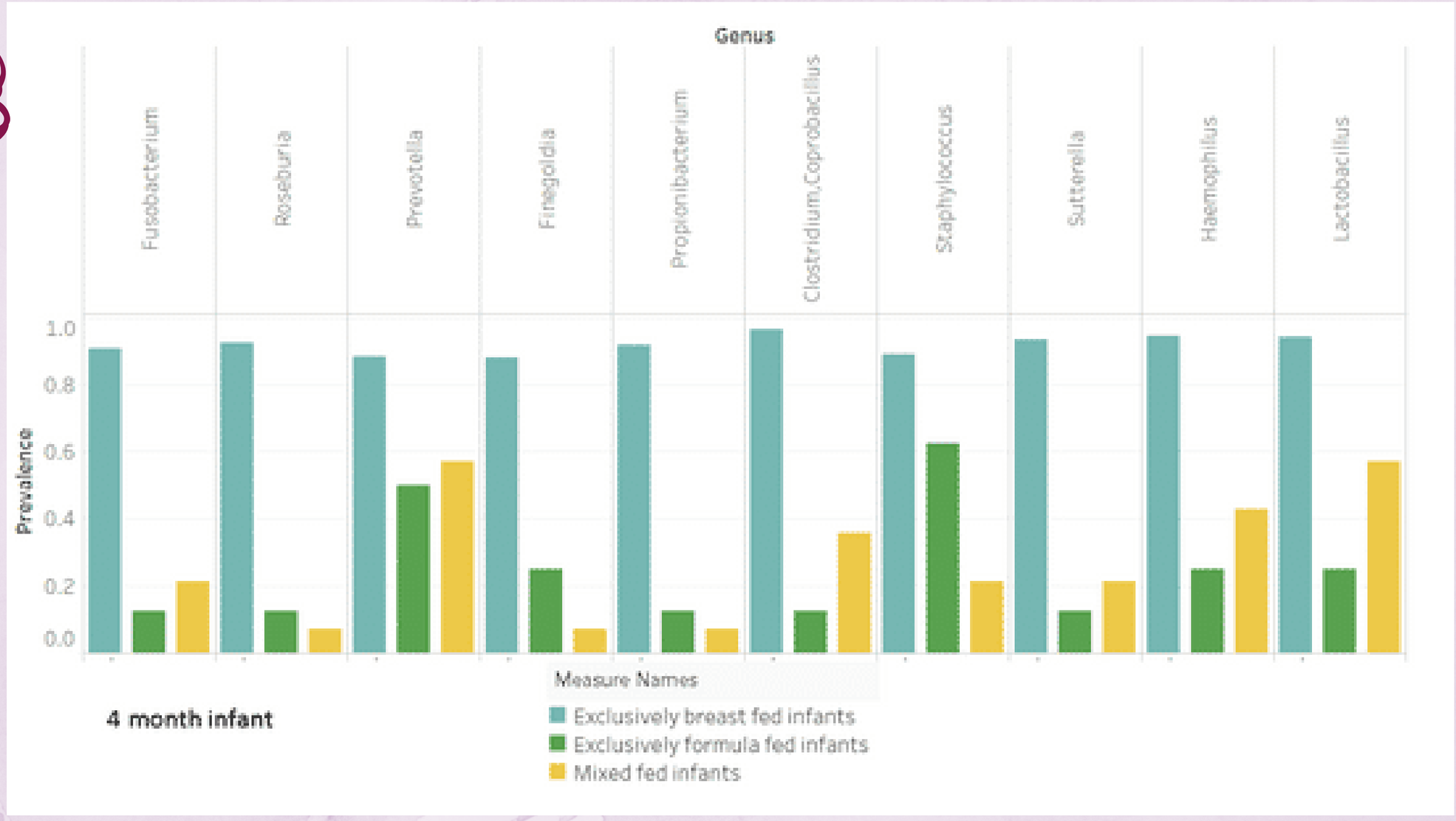
OUR DATA SOURCE:

A PUBLISHED PAPER "DYNAMICS AND STABILIZATION OF THE HUMAN GUT MICROBIOME DURING THE FIRST YEAR OF LIFE"

XGBOOST WAS THE MODEL USED FOR THE MICROBIOME DATA AS WELL AS OUR MOM/BABY DATA.







Shifts in the Microbiome of 4 month old infants:
Breast Fed, Formula Fed and Mixed Fed.

In [1]:
```
from xgboost.sklearn import XGBClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
import xgboost as xgb
from sklearn import model_selection, metrics
from sklearn.model_selection import GridSearchCV
```

In [2]:
```
import pandas as pd
import os
import matplotlib.pyplot as plt
import seaborn as sns; sns.set()
import numpy as np
```

In [4]:
```
df = pd.read_csv(os.path.join("microbiome_dataset_prevalence_
df.head()
```

Out[4]:

	Genus	Month	Delivery_type	Infant_Prevalence	Mother_Preval
0	Acidaminococcus	0	Vaginal	3.0	
1	Acidovorax	0	Vaginal	0.0	
2	Actinobacillus	0	Vaginal	2.0	
3	Actinomyces	0	Vaginal	6.0	
4	Aeromonas	0	Vaginal	0.0	

In [5]:
```
df1 = df[['Month', 'Delivery_type', 'Infant_Prevalence','Mother_Prevalence','
df1.head()
```

Out[5]:

	Month	Delivery_type	Infant_Prevalence	Mother_Prevalence	Genus
0	0	Vaginal	3.0	27	Acidaminococcus
1	0	Vaginal	0.0	1	Acidovorax
2	0	Vaginal	2.0	0	Actinobacillus
3	0	Vaginal	6.0	65	Actinomyces
4	0	Vaginal	0.0	0	Aeromonas

In [6]:
```
one_hot_df1 = pd.get_dummies(df1, prefix=['Delivery_type','Genus'])
one_hot_df1.head()
```

Out[6]:

	Month	Infant_Prevalence	Mother_Prevalence	Delivery_type_C-section	Delivery_type_Vaginal	Genu
0	0	3.0	27	0	1	
1	0	0.0	1	0	1	
2	0	2.0	0	0	1	
3	0	6.0	65	0	1	
4	0	0.0	0	0	1	

5 rows × 127 columns

In [7]:
```
target = one_hot_df1["Month"]
target_names = ["newborn", "4M"]
```

In [7]:
```
target = one_hot_df1["Month"]
target_names = ["newborn", "4M"]
```

In [8]:
```
data = one_hot_df1.drop("Month", axis=1)
feature_names = data.columns
data.head()
```

Out[8]:

	Infant_Prevalence	Mother_Prevalence	Delivery_type_C-section	Delivery_type_Vaginal	Genus_Acidaminococcus	Genus_Acido
0	3.0	27	0	1	1	
1	0.0	1	0	1	0	
2	2.0	0	0	1	0	
3	6.0	65	0	1	0	
4	0.0	0	0	1	0	

5 rows × 126 columns

In [58]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(data, target,  test_size=0.3,random_state=72)
```

In [59]:
```
model = XGBClassifier(n_estimators=1000,colsample_bylevel=0.8, gamma=10)
model.fit(X_train, y_train)
print(model)
```

```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=0.8,
              colsample_bynode=0.8, colsample_bytree=1, gamma=10,
              learning_rate=0.1, max_delta_step=0, max_depth=3,
              min_child_weight=1, missing=None, n_estimators=1000, n_jobs=1,
              nthread=None, objective='binary:logistic', random_state=0,
              reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
              silent=None, subsample=1, verbosity=1)
```

In [60]:
```
y_pred = model.predict(X_test)
predictions =[round(value) for value in y_pred]
```

In [61]:
```
accuracy = accuracy_score(y_test, predictions)
print("Accuracy: %.2f%%" %(accuracy*100))
```

```
Accuracy: 78.23%
```

Machine learning: Pandas notebook with the microbiome data, Accuracy of 78.23%

How did our Project Develop?

HOW WE DID IT AND WHAT WE LEARNED.

LISTEN MORE THAN YOU SPEAK.

Picking out a topic and then data to match the topic. Once you find the data, let the data speak and help you tell the story.

DON'T BE AFRAID.

The learning process can be difficult. Document everything. Break every task into the smallest pieces so that tasks can be completed. But also, don't be afraid to try.

WE DEVELOPED AN APPLICATION

The application took many paths before it's final form was realized. The data available needed to tell the story and how we would use it.

WE ANALYZED SOURCES AND READ DOCUMENTATION

Data analytics means reading whatever is available to learn more about your topic. Then reading everything you can find on how to get your data.

WE EXTRACTED, TRANSFORMED AND LOADED DATA

No data job is really complete without this step

WE DOCUMENTED EVERYTHING.

Anything we reviewed, used, considered was documented as a part of the learning process. This documentation was used to help guide us through the testing and building process.

WE BUILT A DATABASE

The database was loaded with all the cleaned data we estimated we would need to use to complete the project, machine learning and visualizations.

WE BUILT VISUALS

Analysis of the data through visualization is the best way to get a message through to not just the end viewer but also your team mates.

WE BUILT MODELS

We built machine learning models and tested and trained them based on all our learning. We started with linear regression and realized we had to change. We used Random Forest, XGboost and more

SOMETHING TO
THINK ABOUT

"The Greatest Teacher Failure is ..."

YODA

If we had not struggled, We would not have learned as much.

TEAM