MET CS 677 Sprint 2 – Term Project Output

Sushant Khot

04/29/2021

Topic Selected: Fetal Health Classification

================================

# Introduction

================================

Context: I decided to choose this dataset having experienced the birth of my premature kid who is doing great today.

I wanted to examine and study the dataset to understand what analysis we can do on the fetal health data.

Tasks performed:

- Examine which features have strong correlation with Class label.
- Perform Classification Analysis and build models using :

1. KNN Classifier (find optimal k and re-run to get the final accuracy)
2. Logistic Regression classifier
3. Naive Bayesian
4. Decision Tree
5. Random Forest

- I calculated and discussed Performance Metrics for these classifiers by preparing Confusion Matrix to look at how our prediction models perform.
- I have Split Data into Training and Testing to verify the models built (50/50).
- I also tried to visualize the dataset and look for features that can help for classification, outliers, correlation etc.

Some Questions that I tried to answer at the end of Project and analysis:

1. What features in different Visualizations show us trends and help classify a fetus.

2. Compare and build insights on different features and their correlation with the Class Label.

3. Which Model best predicts the Health of a fetus?

4. Compare all classifiers listed above and discuss our findings using confusion matrix.

Data: This dataset contains 2126 records of features extracted from Cardiotocogram exams, which were then classified by three expert obstetricians into 3 classes:

- Normal
- Suspect
- Pathological

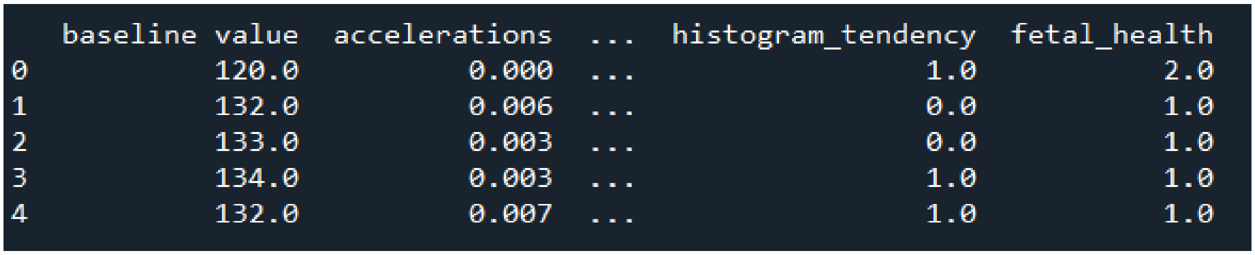
I have combined "Suspect" and "Pathological" classes into one class called "Abnormal" and Normal will stay as "Normal".

# ===

#   Importing the Fetal Heath dataset in pandas dataframe

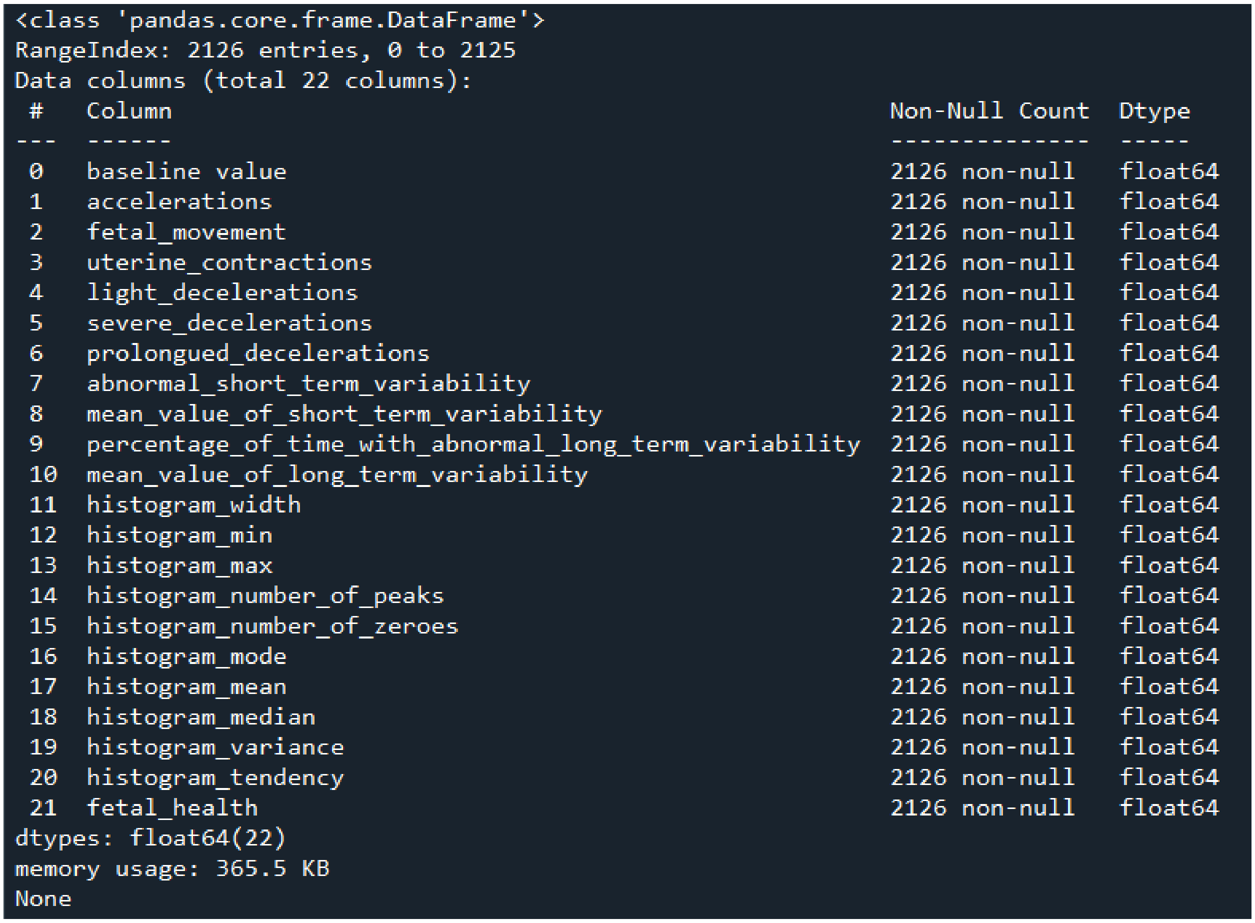
# ===

We will validate the import of data in our pandas df by printing the top 5 or head() of the data frame:



```
   baseline value  accelerations  ...  histogram_tendency  fetal_health
0           120.0          0.000  ...                 1.0           2.0
1           132.0          0.006  ...                 0.0           1.0
2           133.0          0.003  ...                 0.0           1.0
3           134.0          0.003  ...                 1.0           1.0
4           132.0          0.007  ...                 1.0           1.0
```

We will check the info of the data to check if there are any null values.



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2126 entries, 0 to 2125
Data columns (total 22 columns):
 #   Column                                                  Non-Null Count  Dtype  
---  ------                                                  --------------  -----  
 0   baseline value                                          2126 non-null   float64
 1   accelerations                                           2126 non-null   float64
 2   fetal_movement                                          2126 non-null   float64
 3   uterine_contractions                                    2126 non-null   float64
 4   light_decelerations                                     2126 non-null   float64
 5   severe_decelerations                                    2126 non-null   float64
 6   prolongued_decelerations                                2126 non-null   float64
 7   abnormal_short_term_variability                         2126 non-null   float64
 8   mean_value_of_short_term_variability                    2126 non-null   float64
 9   percentage_of_time_with_abnormal_long_term_variability  2126 non-null   float64
 10  mean_value_of_long_term_variability                     2126 non-null   float64
 11  histogram_width                                         2126 non-null   float64
 12  histogram_min                                           2126 non-null   float64
 13  histogram_max                                           2126 non-null   float64
 14  histogram_number_of_peaks                               2126 non-null   float64
 15  histogram_number_of_zeroes                              2126 non-null   float64
 16  histogram_mode                                          2126 non-null   float64
 17  histogram_mean                                          2126 non-null   float64
 18  histogram_median                                        2126 non-null   float64
 19  histogram_variance                                      2126 non-null   float64
 20  histogram_tendency                                      2126 non-null   float64
 21  fetal_health                                            2126 non-null   float64
dtypes: float64(22)
memory usage: 365.5 KB
None
```

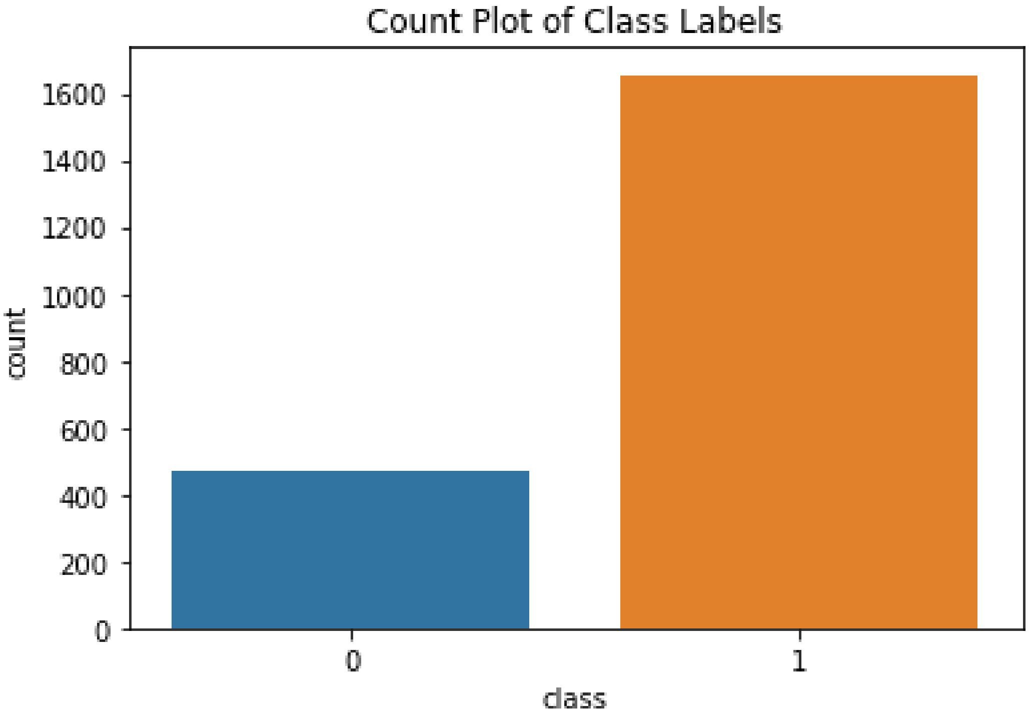
We see that there are **2126** total rows in the data set and none of them are of null value.

Hence our dataset for fetal health is pretty clean and does not need any null value fixes.

We will add the "class" labels to identify the 3 types of fetal health: 1 = "Normal", 2 = "Suspect" and 3 = "Pathological" and combine the labels in 2 groups, "Normal" - these labels are assigned and "Abnormal" - everything else.

Count Plot of Class Labels:

We will look the count plot to understand how our dataset is distributed in terms of each class value.



It seems our dataset has 1655 “Normal” class data and 471 "Abnormal" class data.

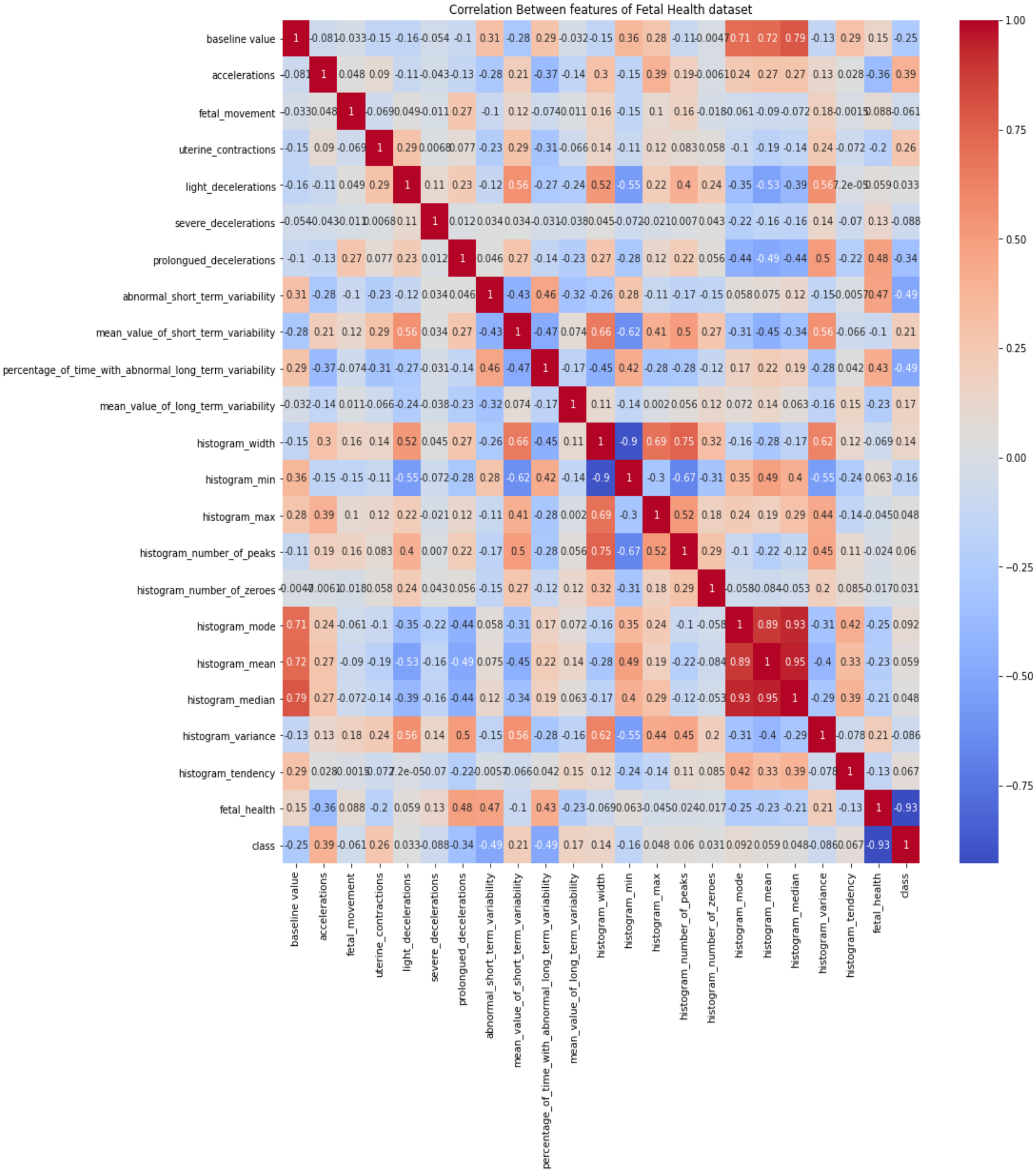
# ==

#   Correlation Matrix of features

# ==

The plot shows top 4 features "Prolonged Deceleration", "Abnormal Short Term Variability", "Percentage of time with abnormal long term variability" and "Accelerations" are highly correlated to "Fetal Health", in that order.

"Histogram number of zeros", "Histogram number of peaks", "Histogram Max" and "Light Decelerations" are least correlated to "Fetal Health", in that order.


Correlation Between features of Fetal Health dataset
baseline value 1 -0.081 -0.033 -0.15 -0.16 -0.054 -0.1 0.31 -0.28 0.29 -0.032 -0.15 0.36 0.28 -0.11 -0.0047 0.71 0.72 0.79 -0.13 0.29 0.15 -0.25
accelerations -0.081 1 0.048 0.09 -0.11 -0.043 -0.13 -0.28 0.21 -0.37 -0.14 0.3 -0.15 0.39 0.19 -0.0061 0.24 0.27 0.27 0.13 0.028 -0.36 0.39
fetal_movement -0.033 0.048 1 -0.069 0.049 -0.011 0.27 -0.1 0.12 -0.074 0.011 0.16 -0.15 0.1 0.16 -0.018 -0.061 -0.09 -0.072 0.18 -0.0015 0.088 -0.061
uterine_contractions -0.15 0.09 -0.069 1 0.29 0.0068 0.077 -0.23 0.29 -0.31 -0.066 0.14 -0.11 0.12 0.083 0.058 -0.1 -0.19 -0.14 0.24 -0.072 -0.2 0.26
light_decelerations -0.16 -0.11 0.049 0.29 1 0.11 0.23 -0.12 0.56 -0.27 -0.24 0.52 -0.55 0.22 0.4 0.24 -0.35 -0.53 -0.39 0.56 7.2e-05 0.059 0.033
severe_decelerations -0.054 -0.043 -0.011 0.0068 0.11 1 0.012 0.034 0.034 -0.031 -0.038 0.045 -0.072 -0.021 0.007 0.043 -0.22 -0.16 -0.16 0.14 -0.07 0.13 -0.088
prolongued_decelerations -0.1 -0.13 0.27 0.077 0.23 0.012 1 0.046 0.27 -0.14 -0.23 0.27 -0.28 0.12 0.22 0.056 -0.44 -0.49 -0.44 0.5 -0.22 0.48 -0.34
abnormal_short_term_variability 0.31 -0.28 -0.1 -0.23 -0.12 0.034 0.046 1 -0.43 0.46 -0.32 -0.26 0.28 -0.11 -0.17 -0.15 0.058 0.075 0.12 -0.15 -0.0057 0.47 -0.49
mean_value_of_short_term_variability -0.28 0.21 0.12 0.29 0.56 0.034 0.27 -0.43 1 -0.47 0.074 0.66 -0.62 0.41 0.5 0.27 -0.31 -0.45 -0.34 0.56 -0.066 -0.1 0.21
percentage_of_time_with_abnormal_long_term_variability 0.29 -0.37 -0.074 -0.31 -0.27 -0.031 -0.14 0.46 -0.47 1 -0.17 -0.45 0.42 -0.28 -0.28 -0.12 0.17 0.22 0.19 -0.28 0.042 0.43 -0.49
mean_value_of_long_term_variability -0.032 -0.14 0.011 -0.066 -0.24 -0.038 -0.23 -0.32 0.074 -0.17 1 0.11 -0.14 0.002 0.056 0.12 0.072 0.14 0.063 -0.16 0.15 -0.23 0.17
histogram_width -0.15 0.3 0.16 0.14 0.52 0.045 0.27 -0.26 0.66 -0.45 0.11 1 -0.9 0.69 0.75 0.32 -0.16 -0.28 -0.17 0.62 0.12 -0.069 0.14
histogram_min 0.36 -0.15 -0.15 -0.11 -0.55 -0.072 -0.28 0.28 -0.62 0.42 -0.14 -0.9 1 -0.3 -0.67 -0.31 0.35 0.49 0.4 -0.55 -0.24 0.063 -0.16
histogram_max 0.28 0.39 0.1 0.12 0.22 -0.021 0.12 -0.11 0.41 -0.28 0.002 0.69 -0.3 1 0.52 0.18 0.24 0.19 0.29 0.44 -0.14 -0.045 0.048
histogram_number_of_peaks -0.11 0.19 0.16 0.083 0.4 0.007 0.22 -0.17 0.5 -0.28 0.056 0.75 -0.67 0.52 1 0.29 -0.1 -0.22 -0.12 0.45 0.11 -0.024 0.06
histogram_number_of_zeroes -0.0047 -0.0061 -0.018 0.058 0.24 0.043 0.056 -0.15 0.27 -0.12 0.12 0.32 -0.31 0.18 0.29 1 -0.058 -0.084 -0.053 0.2 0.085 -0.017 0.031
histogram_mode 0.71 0.24 -0.061 -0.1 -0.35 -0.22 -0.44 0.058 -0.31 0.17 0.072 -0.16 0.35 0.24 -0.1 -0.058 1 0.89 0.93 -0.31 0.42 -0.25 0.092
histogram_mean 0.72 0.27 -0.09 -0.19 -0.53 -0.16 -0.49 0.075 -0.45 0.22 0.14 -0.28 0.49 0.19 -0.22 -0.084 0.89 1 0.95 -0.4 0.33 -0.23 0.059
histogram_median 0.79 0.27 -0.072 -0.14 -0.39 -0.16 -0.44 0.12 -0.34 0.19 0.063 -0.17 0.4 0.29 -0.12 -0.053 0.93 0.95 1 -0.29 0.39 -0.21 0.048
histogram_variance -0.13 0.13 0.18 0.24 0.56 0.14 0.5 -0.15 0.56 -0.28 -0.16 0.62 -0.55 0.44 0.45 0.2 -0.31 -0.4 -0.29 1 -0.078 0.21 -0.086
histogram_tendency 0.29 0.028 -0.0015 -0.072 7.2e-05 -0.07 -0.22 -0.0057 -0.066 0.042 0.15 0.12 -0.24 -0.14 0.11 0.085 0.42 0.33 0.39 -0.078 1 -0.13 0.067
fetal_health 0.15 -0.36 0.088 -0.2 0.059 0.13 0.48 0.47 -0.1 0.43 -0.23 -0.069 0.063 -0.045 -0.024 -0.017 -0.25 -0.23 -0.21 0.21 -0.13 1 -0.93
class -0.25 0.39 -0.061 0.26 0.033 -0.088 -0.34 -0.49 0.21 -0.49 0.17 0.14 -0.16 0.048 0.06 0.031 0.092 0.059 0.048 -0.086 0.067 -0.93 1
baseline value
accelerations
fetal_movement
uterine_contractions
light_decelerations
severe_decelerations
prolongued_decelerations
abnormal_short_term_variability
mean_value_of_short_term_variability
percentage_of_time_with_abnormal_long_term_variability
mean_value_of_long_term_variability
histogram_width
histogram_min
histogram_max
histogram_number_of_peaks
histogram_number_of_zeroes
histogram_mode
histogram_mean
histogram_median
histogram_variance
histogram_tendency
fetal_health
class
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75

# ==

**#	Feature Selection using pearson's correlation**

# ==

We will use pearson's correlation feature selection method to select the top 10 features for our analysis.

The top 10 features selected are as follows:

- baseline value
- accelerations
- uterine_contractions
- prolongued_decelerations
- abnormal_short_term_variability
- mean_value_of_short_term_variability
- percentage_of_time_with_abnormal_long_term_variability
- mean_value_of_long_term_variability
- histogram_width
- histogram_min

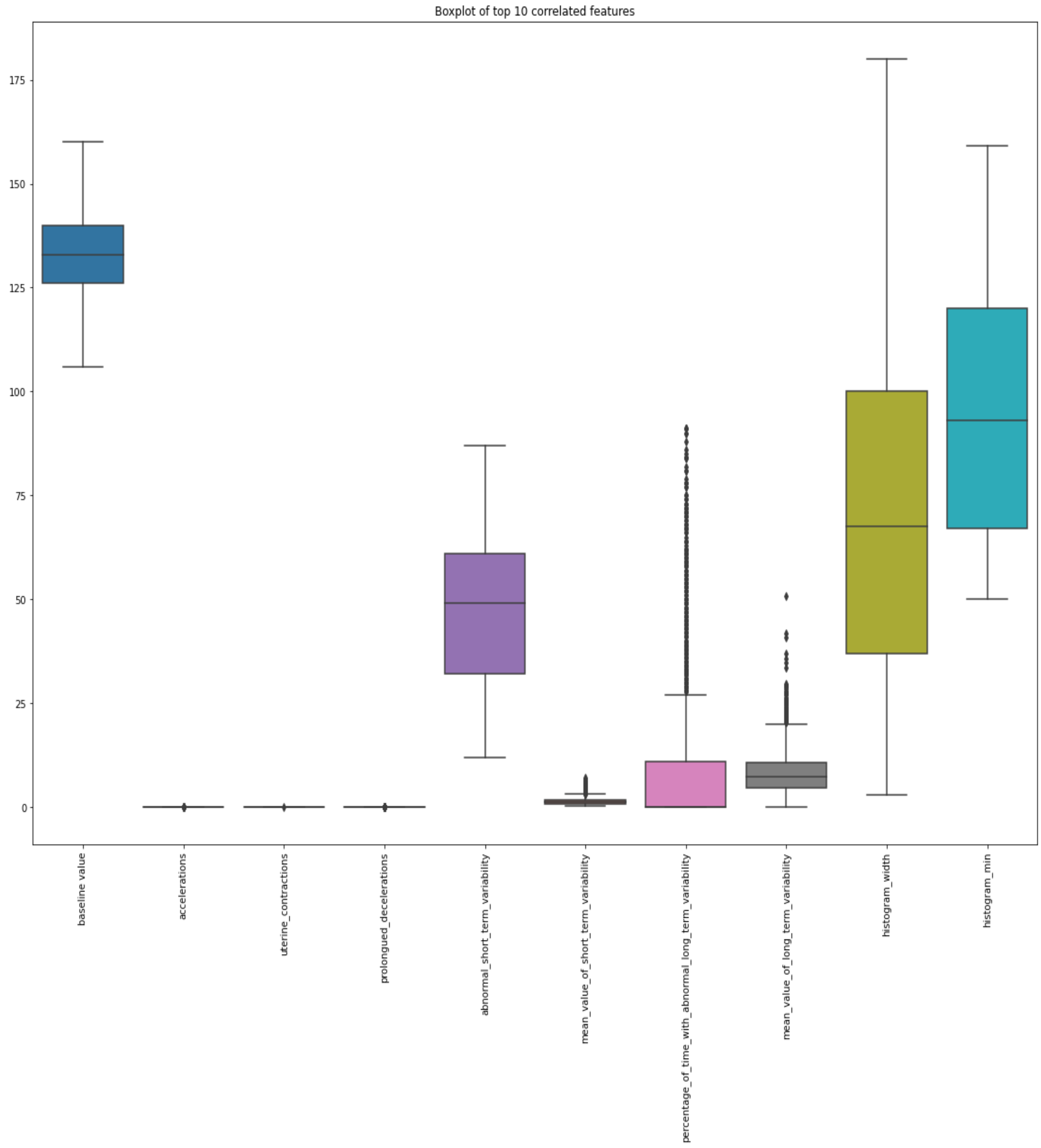
Boxplot for outliers check:

We will check if there are any outliers in the dataset using box plots.

Based on the plot below, we do see many features have outliers specifically "Percentage of time with abnormal long term variability", "Mean value of long term variability" and "Mean value of short term variability".

However since it’s a medical dataset, the outcome of the CTG report is unlikely to have any data entry.

We will hence not remove any outliers and continue our analysis on the dataset.

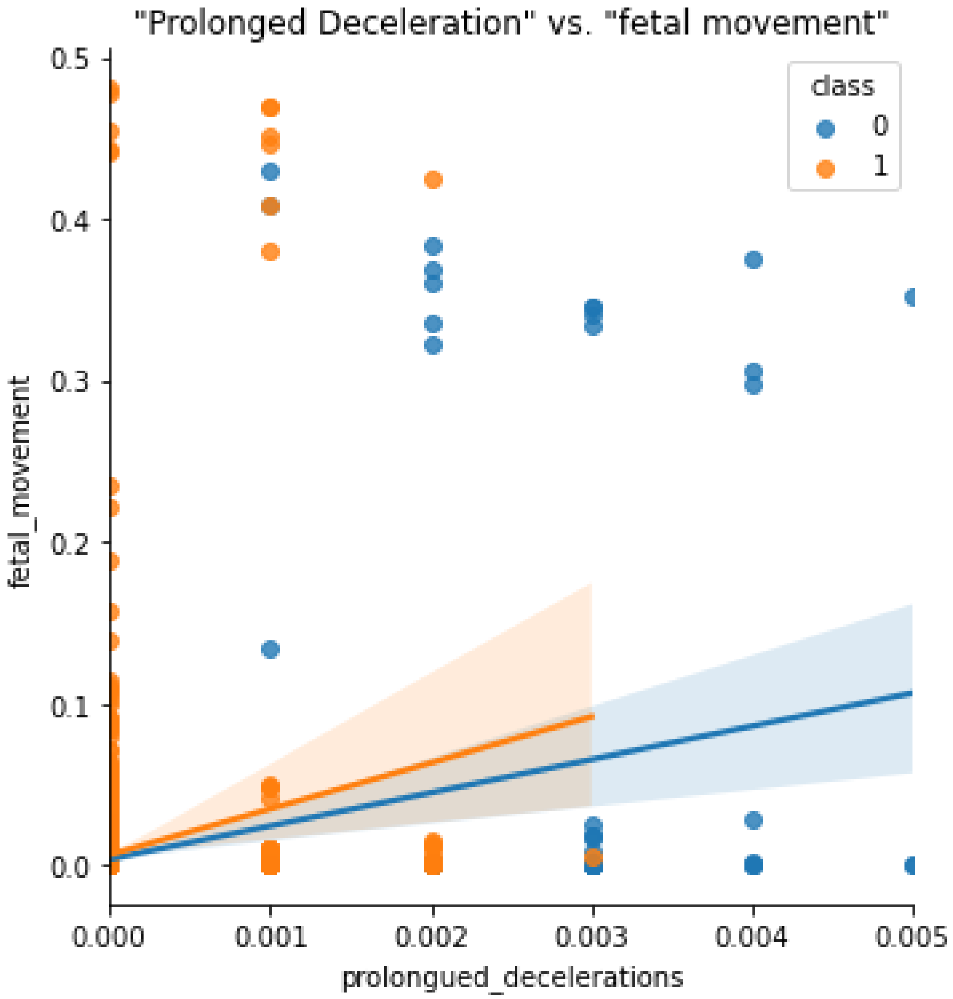


Boxplot of top 10 correlated features
175
150
125
100
75
50
25
0
baseline value
accelerations
uterine_contractions
prolongued_decelerations
abnormal_short_term_variability
mean_value_of_short_term_variability
percentage_of_time_with_abnormal_long_term_variability
mean_value_of_long_term_variability
histogram_width
histogram_min

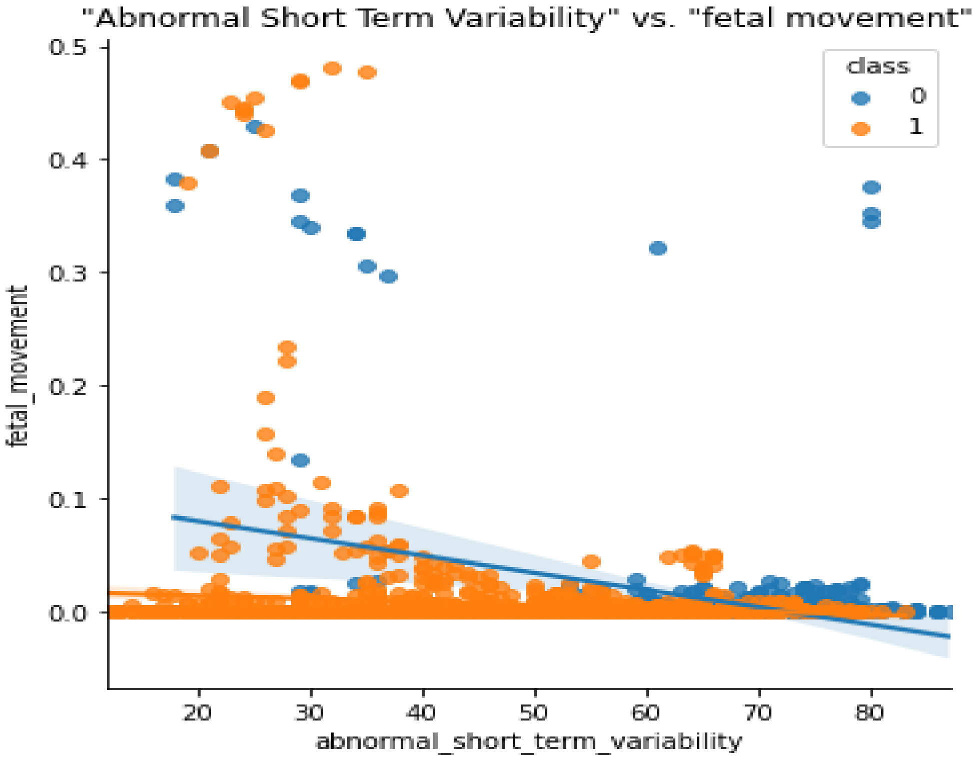
Regression Plots:

Let us look at the regression plots for the top 4 most correlated features with fetal_health vs fetal_movement attribute:

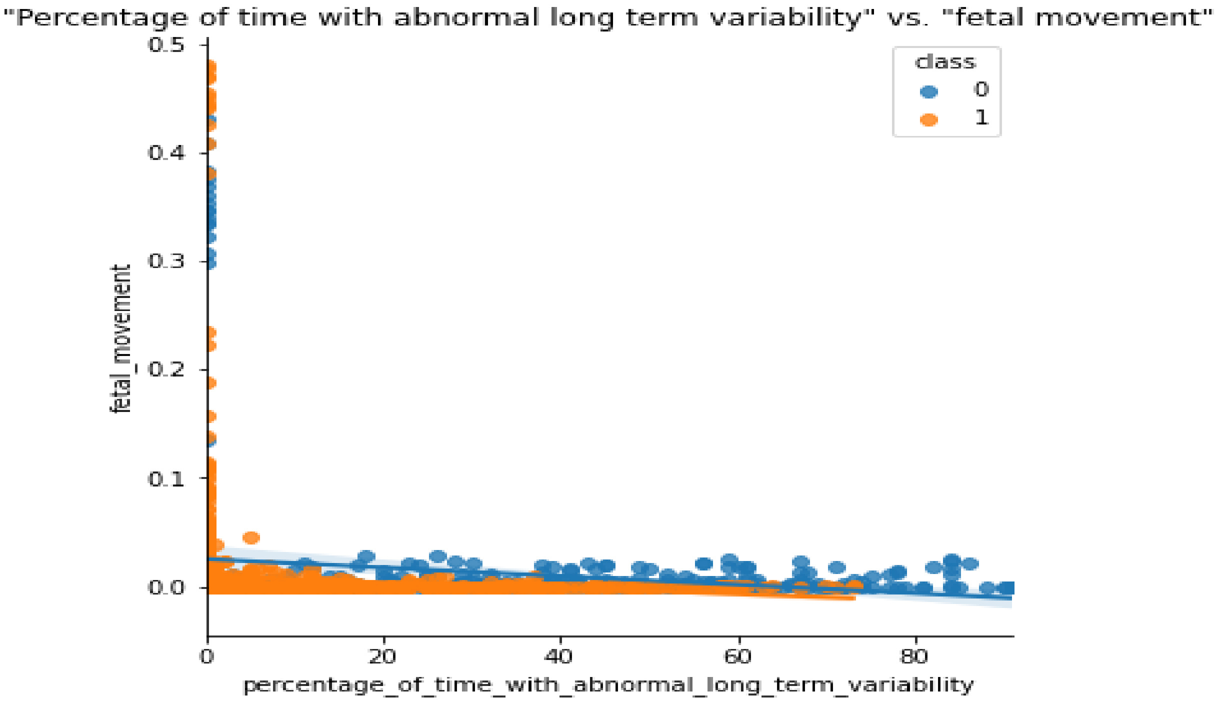
1. "Prolonged Deceleration" vs. "fetal_movement"



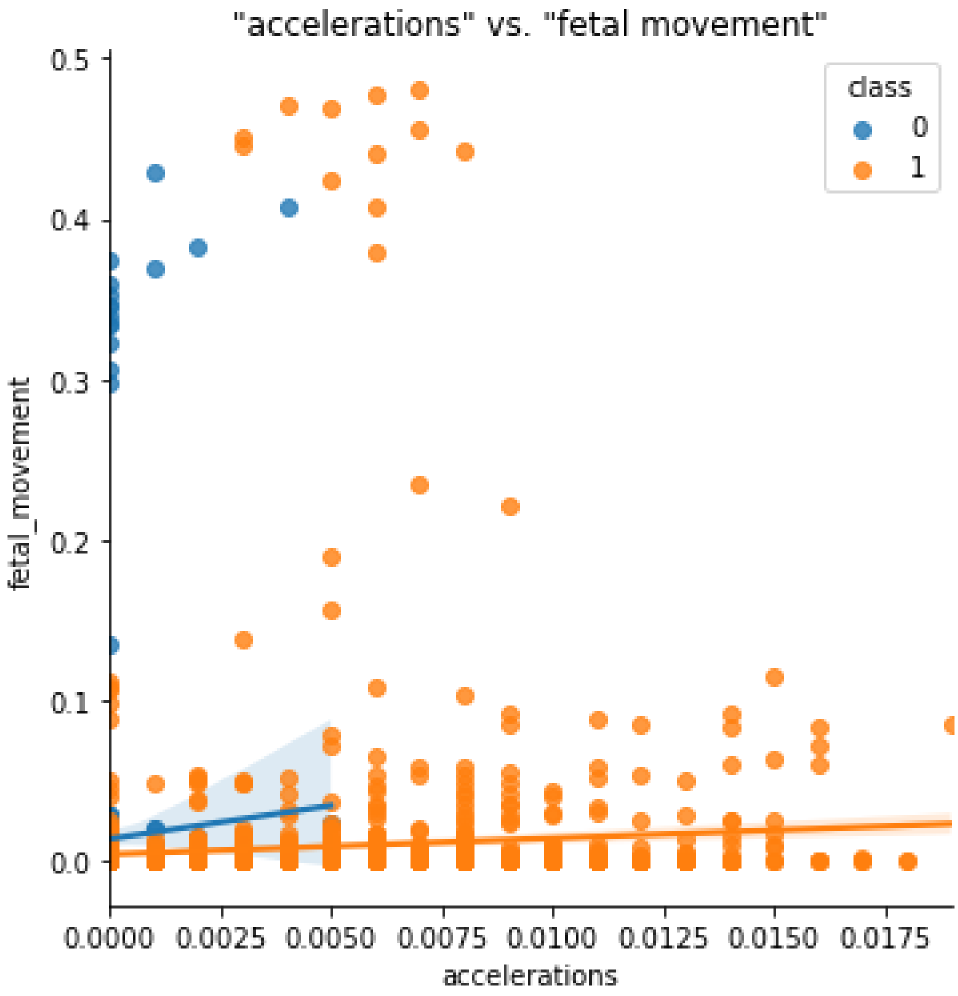
2. "Abnormal Short Term Variability" vs. "fetal_movement"



3. **"Percentage of time with abnormal long term variability" vs. "fetal_movement"**



# 4. "accelerations" vs. "fetal_movement"



We confirm the correlation behavior from these plots.

Also, we can see there are some outliers like we found in our boxplots.

# ===

# Splitting dataset 50/50 train/test

# ===

We will be splitting the dataset 50/50 using train_test_split. This way we will train 50% of the data and test the remaining 50% on it.

I am assuming **Positive** event as Fetal Health Class = “**Normal**” or “1” and **Negative** Event as Fetal Health Class = “**Abnormal**” or "0".

# ==

We will now run some classifier models listed below on the 50/50 train and test data to classify the labels and predict their accuracy.

Models that we will run are as follows:

1. KNN Classifier (find optimal k and re-run to get the final accuracy)

2. Logistic Regression classifier

3. Naive Bayesian

4. Decision Tree

5. Random Forest

# ==

# ==

# k-NN classifier using sklearn library

# ==

- We will take k = 3, 5, 7, 9, 11. Use the same Xtrain and Xtest as before.
- For each k, we will train the k-NN classifier on Xtrain and compute its accuracy for Xtest

We will scale the training and testing data and stored scaled data in X_train_scaled and X_test_scaled respectively

We will now run the KNN classifier for k = 3, 5, 7, 9, 11 and print out the Accuracies for each k to find the optimal k*.

The Accuracy for KNN classifier with k = 3 = **92.0%**

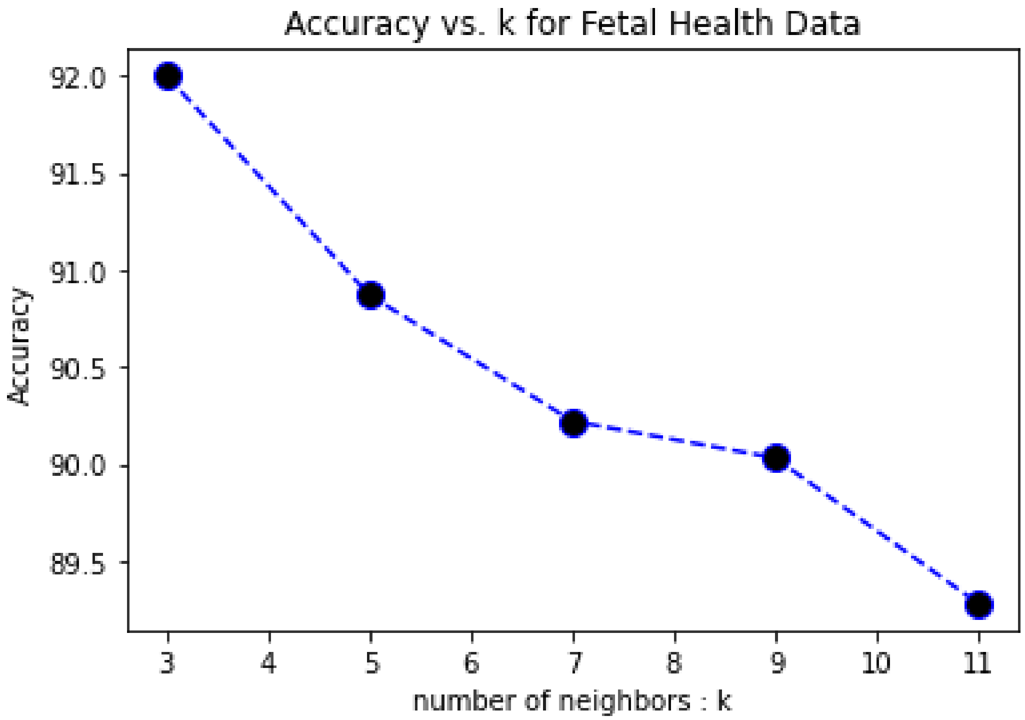
The Accuracy for KNN classifier with k = 5 = **90.87%**

The Accuracy for KNN classifier with k = 7 = **90.22%**

The Accuracy for KNN classifier with k = 9 = **90.03%**

The Accuracy for KNN classifier with k = 11 = **89.28%**

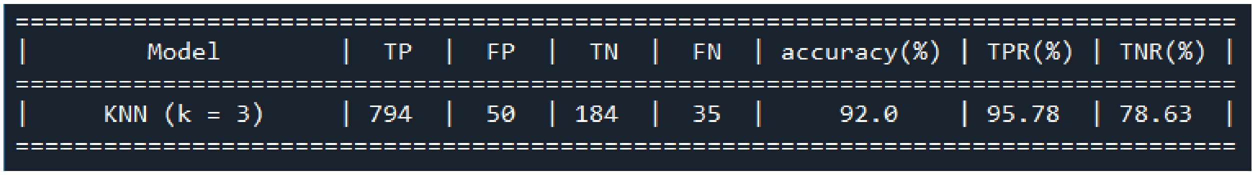
- Plot a graph showing the accuracy.
- On x axis we will plot k and on y-axis we will plot accuracy.
- We will then find the optimal value k* of k.



The optimal value k* is **k = 3**

- We will use the optimal value k* to compute performance measures and summarize them in the table.
-

The Accuracy using KNN (k = 3) is = **92.0%**



```
=================================================================================
|        Model          |  TP  |  FP  |  TN  |  FN  | accuracy(%) | TPR(%) | TNR(%) |
=================================================================================
|     KNN (k = 3)       | 794  |  50  | 184  |  35  |    92.0     | 95.78  | 78.63  |
=================================================================================
```

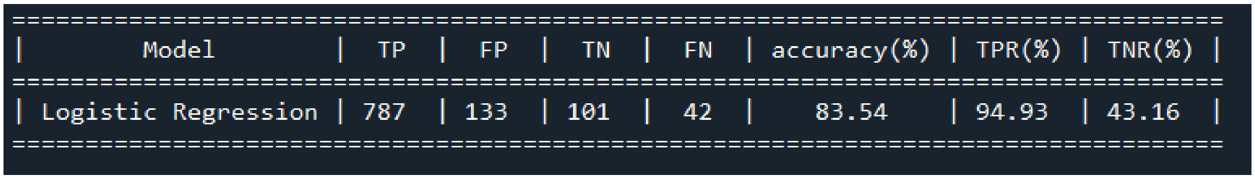
# ==

#   Logistic Regression classifier using sklearn library

# ==

- We will use the same Xtrain and Xtest as before and train our logistic regression classifier on Xtrain and compute its accuracy for Xtest

The Accuracy using Logistic Regression is = **83.54%**



```
=================================================================================
|        Model         |  TP  |  FP  |  TN  |  FN  | accuracy(%) | TPR(%) | TNR(%) |
=================================================================================
| Logistic Regression | 787  | 133  | 101  |  42  |    83.54    | 94.93  | 43.16  |
=================================================================================
```

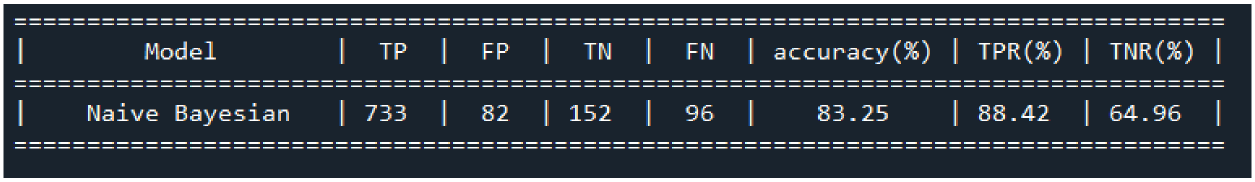
# ===============================

#   Naive Bayesian

# ===============================

- We will run the classifier on our 50/50 dataset, train NB on Xtrain and predict class labels in Xtest.
- We will find out the accuracy and compute the confusion matrix

The Accuracy using Naive Bayesian is = **83.25%**



```
=================================================================================
|        Model         |  TP  |  FP  |  TN  |  FN  | accuracy(%) | TPR(%) | TNR(%) |
=================================================================================
|    Naive Bayesian    | 733  |  82  | 152  |  96  |    83.25    | 88.42  | 64.96  |
=================================================================================
```

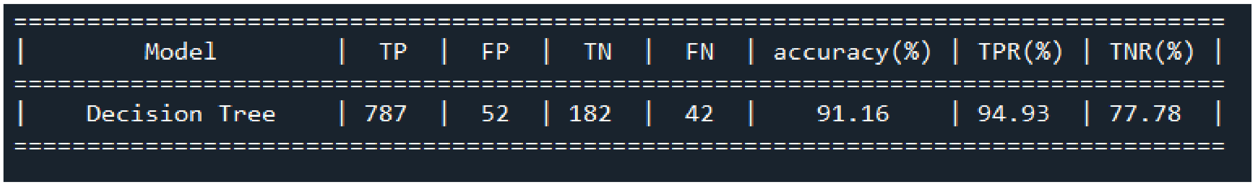
================================

# **Decision Tree**

================================

- We will run the classifier on our 50/50 dataset, train Decision tree on Xtrain and predict class labels in Xtest.
- We will find out the accuracy and compute the confusion matrix

The Accuracy using Decision Tree is = **91.16%**



```
=================================================================================
|        Model         |  TP  |  FP  |  TN  |  FN  | accuracy(%) | TPR(%) | TNR(%) |
=================================================================================
|    Decision Tree     | 787  |  52  | 182  |  42  |    91.16    | 94.93  | 77.78  |
=================================================================================
```

================================

# **Random Forest**

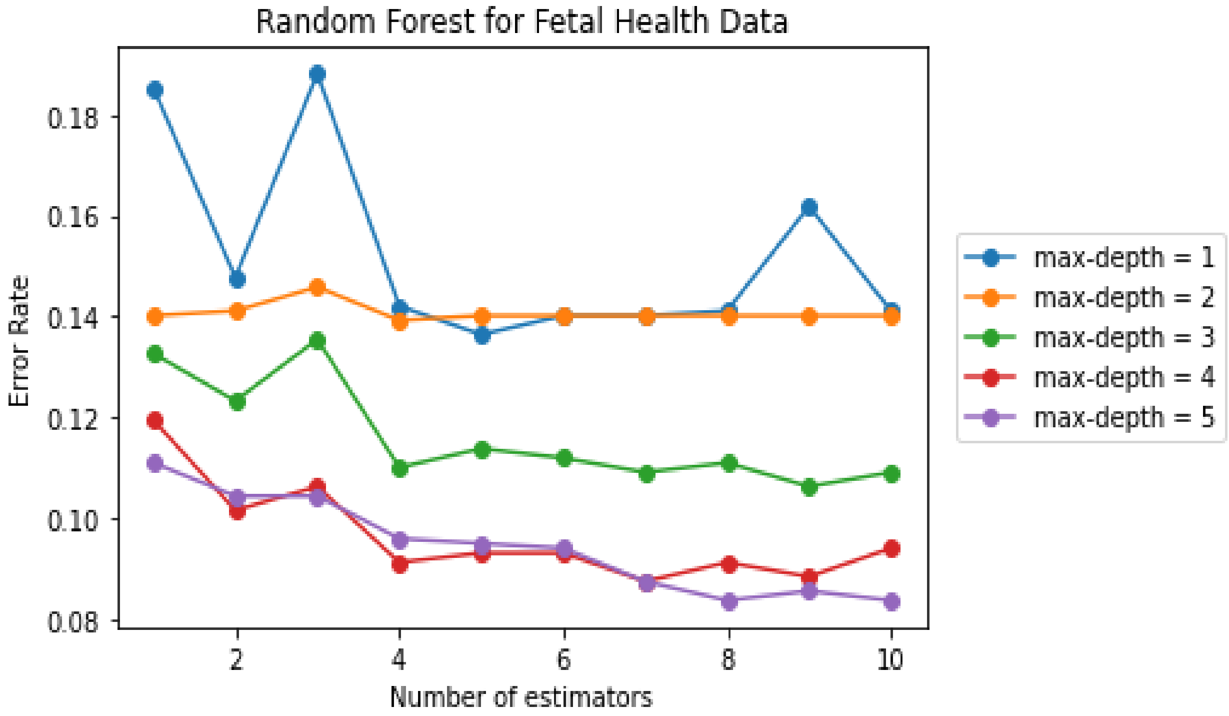
================================

- We will take N = 1,...,10 and d = 1, 2,...,5.
- For each value of N and d, we will split our data into Xtrain and Xtest,
- We will construct a random tree classifier (use "entropy" as splitting criteria - this is the default)
- Finally, we will train our classifier on Xtrain and compute the error rate for Xtest.

We will create a dictionary "error_rates_dict_RF", that will store the respective 10 error rates for each estimator. In turn these 10 error rates will be stored with max-depth value key name.

This dictionary setup will help us to plot the data accordingly.

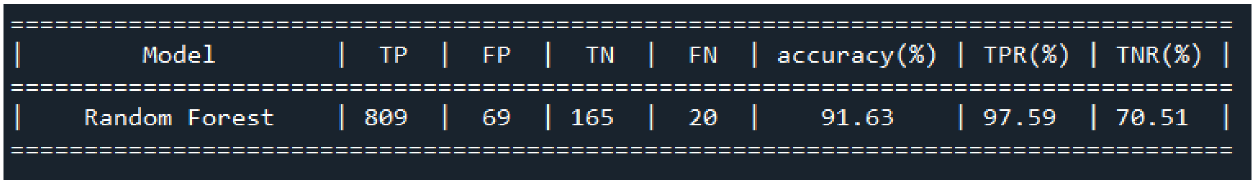
- We will now plot the error rates and find the best combination of N and d.
- We will then calculate the accuracy for the best combination of N and k.
- We will then compute the confusion matrix using the best combination of N and d.



From the plot we can see that our best number of estimators N = 8 and max-depth d = 5

Accuracy and Confusion Matrix using Random Forest with best N = 8 and d = 5:

The Accuracy using Random Forest with best N = 8 and d = 5 is = **91.63%**



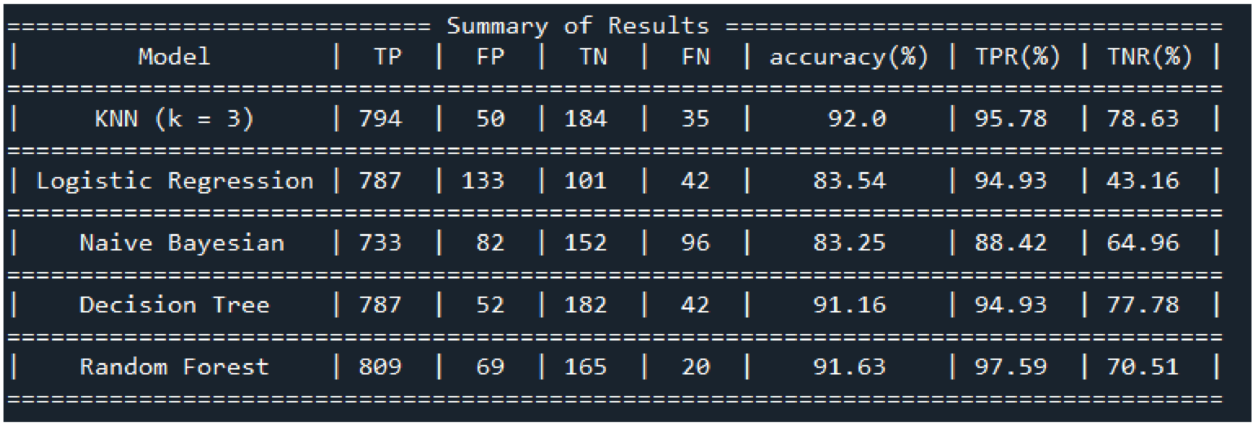
```
===================================================================================
|        Model        |  TP  |  FP  |  TN  |  FN  | accuracy(%) | TPR(%) | TNR(%) |
===================================================================================
|    Random Forest    | 809  |  69  | 165  |  20  |    91.63    | 97.59  | 70.51  |
===================================================================================
```

==

# Confusion Matrix Summary

==

- We will now summarize our results for all the classifiers in a table and discuss our findings.



```
============================ Summary of Results ================================
|        Model         |  TP  |  FP  |  TN  |  FN  | accuracy(%) | TPR(%) | TNR(%) |
==================================================================================
|     KNN (k = 3)      | 794  |  50  | 184  |  35  |    92.0     | 95.78  | 78.63  |
==================================================================================
| Logistic Regression  | 787  | 133  | 101  |  42  |    83.54    | 94.93  | 43.16  |
==================================================================================
|    Naive Bayesian    | 733  |  82  | 152  |  96  |    83.25    | 88.42  | 64.96  |
==================================================================================
|    Decision Tree     | 787  |  52  | 182  |  42  |    91.16    | 94.93  | 77.78  |
==================================================================================
|    Random Forest     | 809  |  69  | 165  |  20  |    91.63    | 97.59  | 70.51  |
==================================================================================
```

Summary of Confusion Matrix results:

Here, Positive event is "Normal"" / class 1 of Fetal health. Negative event is "Abnormal" or class 0 of Fetal Health.

KNN (k = 3) classifier gave us the **best** overall accuracy of **92%** which is close to **Random Forest** and **Decision tree**.

It also predicted the most True Negatives among other classifiers closely followed by Decision Tree classifier.

It's True Negative Rate TNR = **78.63%** which is close to TNR of Decision Tree = **77.78%**

In Terms of correctly predicting the positive event, we see that **Random Forest** has predicted the most true Positives with TPR = **97.59 %,** followed by **KNN (k = 3).**

Decision Tree and **Logistic regression** predicted True Positives with same accuracy.

Logistic Regression classifier has predicted the most False Positives which means that it has classified many "Abnormal" class fetal health as "Normal". However Logistic regression did a good job at predicting the "Normal" class Fetal Health.

Naive Bayesian classifier on the other hand has predicted the most False negatives which means it predicted many "Normal" class Fetal health as "Abnormal". The overall accuracy of Naive Bayesian classifier is the lowest with **83.25%.**

Logistic regression classifier also has a low overall accuracy of **83.54%.**

Overall based on Accuracy, KNN (k = 3), Random Forest and Decision tree seemed to have performed well.

Random Forest is able to classify more "Normal" class fetal health correctly.

KNN (k = 3) and Decision Tree have predicted more "Abnormal" class Fetal health correctly.

# ============================

#   Conclusion

# ============================

We started with analyzing our Fetal health dataset and it was pretty clean dataset to begin with and we did not have to do any pre-processing activities.

Some features did have outliers, but we decided not to remove them as the medical data entry errors would be minimum and we went ahead without removing any outliers.

We created various visualizations to help support our analysis and gather more insights on the features present in the dataset.

For e.g. **Count plot** gave us an idea on distribution of our dataset based on class label.

Correlation Matrix provided us the top most correlated and non-correlated features with Fetal Health

Box plots provided us with insights on data distribution within each feature.

Regression plots provided us with correlation and distribution of features around the 2 class labels "Normal" and "Abnormal" fetal health and fetal movement data.

We split our dataset 50/50 into training and testing and ran through multiple classifiers on the data.

We documented the statistics in terms of classifier Accuracy to correctly classify the Fetal health and discussed the confusion matrix.

Finally based on the confusion matrix we concluded that based on Overall Accuracy, KNN (k = 3), Random Forest and Decision tree seemed to have performed well.

Random Forest is able to classify more "Normal" class fetal health correctly.

KNN (k = 3) and Decision Tree have predicted more "Abnormal" class Fetal health correctly.