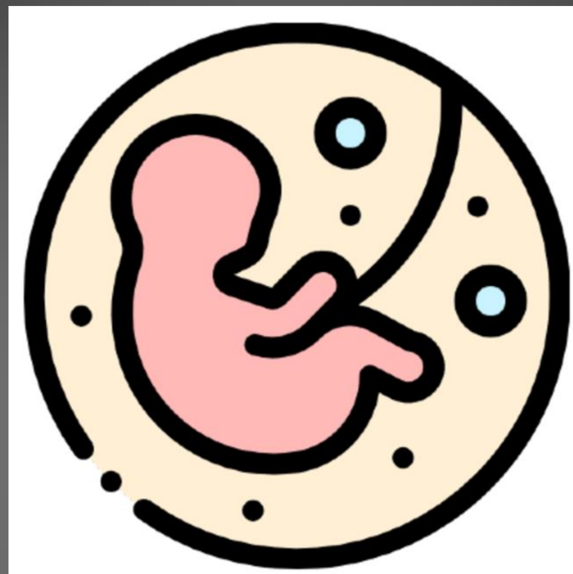

Fetal Health Classification

Student Name : Sathya Priya Mahendran
Student ID : Q15817059
Course Title : Data Analytics and Visualization
Course Leader : Dr Drishty Sobnath
Date : 01/09/2022

Contents

INTRODUCTION

Data science is an interdisciplinary arena intensive on mining information from characteristically big files and smearing the understanding and perceptions from that data to resolve complications in a varied assortment of application areas. Creation intellect of Big Data is the sphere of Data Analytics. There are countless implements and modus operandi which are arrayed in mandate to bring together, make over, cleanse, sort, and transfigure data into straightforwardly logical data visualization and reportage designs. Data visualization is the repetition of interpreting info into a pictorial framework, such as a map or graph, to create data at ease for the humanoid common sense to apprehend and tug perceptions from. The key goal line of data visualization is to brand it tranquil to ascertain outlines, drifts and outliers in big data.

Pregnancy is the cheeriest era in a female's lifetime. During pregnancy, a mommy must really take care of her healthiness with abundant carefulness, since she is carrying a baby. To screen fetal growing and improvement, numerous tests are recommended each trimester. Fetal and infant health consequences are significant methods of the complete wellbeing of a populace and of the feature of condition care services for moms and their offspring.

Decrease of adolescent impermanence is reproduced in numerous of the United Nations' Sustainable Development Goals and means to be crucial display of mortal growth. The UN anticipates that by 2030, nations close unnecessary losses of infants and kids more than 5 years of age, with every countries pointing to lessen lower than-5 impermanence to at least as low as 25 per 1,000 live births. Similar to concept of child mortality is of sequence parental impermanence, which books for 295 000 deaths during and following prenatal period and giving birth (as of 2017). The massive mainstream of these losses (94%) happened in low-resource situations, and most could have been prohibited.

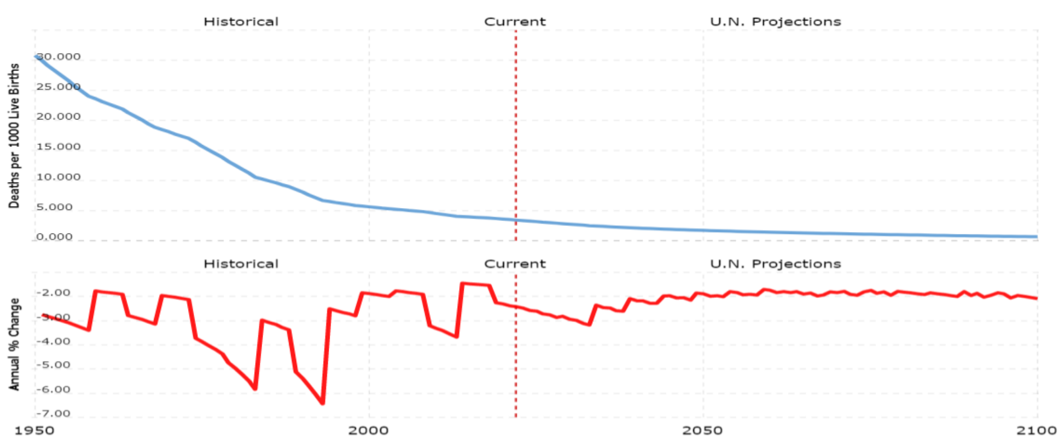

Figure 1: U.K. Infant Mortality Rate 1950-2022

Cardiotocogram (CTG) is the greatly advantageous in the medical repetitive assessment of the foremost method to identify condition of the fetus. It is a guileless and budget reachable choice to measure fetus wellbeing, permitting healthcare authorities to yield exploit in mandate to preclude adolescent and parental impermanence. The kit mechanism by transfer ultrasound pulsations and understanding its retort, thus molting light on fetal heart rate (FHR), fetal movements, uterine contractions and further.

Scope of this Project

The foremost aim of this project is to perform Data Analytics and Data Visualization with the heath dataset. Following steps are involved in the project:

1) **Data collection** of heath related dataset from the open resources such as Kaggle, Google Dataset search, Data.gov.uk, WHO, Dataverse, Office for National Statistics, etc.,

2) **Data preparation** is the method of cleaning and transmuting raw records erstwhile to dealing out and exploration. It is an imperative phase aforementioned to handling and habitually encompasses reformatting facts, constructing rectifications to statistics and the coalescing of data sets to enhance data.

3) **Data exploration**, as well-known as exploratory data analysis, be responsible for a customary of meek tools to succeed an uncomplicated considerate of a dataset. The domino effect of data exploration can be tremendously valuable in clutching the configuration of the data, the scattering of the standards, being there of great ideals, and interrelationships contained by the dataset.

4) **Data modeling** refers to crafting a graphical exemplification of a complete statistics system or roughly of its slices to connect the affiliations among data points and structures. **Data visualization** is actual worthwhile in the present day as corporations are engendering and accumulating giant data works. It can benefit them to blow the whistle on out of sight trinkets from data, which are worthy for progression.

5) **Data evaluation** is to analytically estimate your outcomes and relate your verdicts to further alike studies.

Methodologies

✓ Data Collection

The fetal health dataset is collected from the open platform named Kaggle given by Larxel. This dataset comprises of 2126 records of data mined from Cardiotocogram examinations, completely of numerical values, which existed at that point categorized by 3 proficient obstetricians into three cases: Normal, Suspect and Pathological. These 3 cases come under fetal health target variable. The overall volume and proportion of normal, suspect and pathological data in the fetus wellbeing taxonomy dataset is shown in Figure 2. In the coding portion, normal, suspect, and pathological are replaced as 1.0, 2.0, and 3.0, correspondingly.

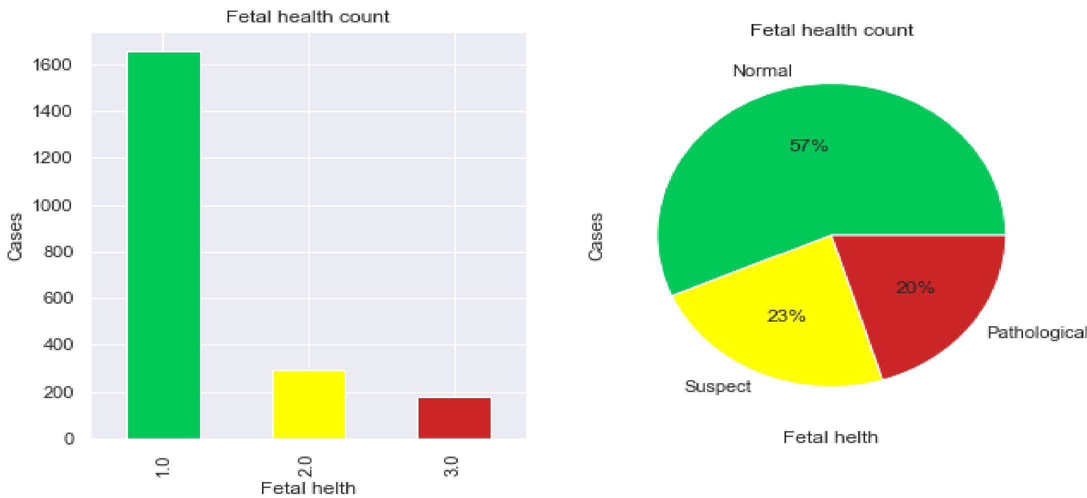

Figure 2: Fetal Health Classification

The dataset was available in CSV format, converted and understood into table format by using Python programming language.

```
#Loading dataset
my_dataset = pd.read_csv(r'C:\Users\sarav\Documents\Sathya_AE2_project\fetal_health.csv')
my_dataset
```

Figure 3: Loading Dataset

Field Variables	Field Description	Type of Variable	No of Categories	Unique Values (Range)	Count
Baseline Value	Baseline Fetal Heart Rate (FHR) beats per minute	Independent	48	106.0 - 160.0	2126
Accelerations	Number of accelerations per second	Independent	20	0.0 - 0.019	2126
Fetal Movement	Number of fetal movements per second	Independent	102	0.0 - 0.481	2126
Uterine Contractions	Number of uterine contractions per second	Independent	16	0.0 - 0.015	2126
Light Decelerations	Number of LDs per second	Independent	16	0.0 - 0.015	2126
Severe Decelerations	Number of SDs per second	Independent	2	0.0 0.001	2119 7
Prolongued Decelerations	Number of PDs per second	Independent	6	0.0 - 0.005	2126
Abnormal Short Term Variability	Percentage of time with abnormal short term variability	Independent	75	12.0 - 87.0	2126
Mean Value of Short Term Variability	Mean value of short term variability	Independent	57	0.2 - 7.0	2126
Percentage of Time with Abnormal Long Term Variability	Percentage of time with abnormal long term variability	Independent	87	77.0 -186.0	2126
Mean Value of Long Term Variability	Mean value of long term variability	Independent	249	0.0 - 50.70	2126
Histogram Width	Width of the histogram made using all values from a record	Independent	154	3.0 - 180.0	2126
Histogram Min	Histogram minimum value (low frequency)	Independent	109	50.0 - 159.0	2126
Histogram Max	Histogram maximum value (high frequency)	Independent	86	122.0 - 238.0	2126
Histogram Number of Peaks	Number of peaks in the exam histogram	Independent	18	0.0 - 18.0	2126
Histogram Number of Zeroes	Number of zeroes in the exam histogram	Independent	9	0.0 - 10.0	2126
Histogram Mode	Histogram mode	Independent	88	60.0 - 187.0	2126
Histogram Mean	Histogram mean	Independent	103	73.0 - 182.0	2126
Histogram Median	Histogram median	Independent	95	77.0 - 186.0	2126
Histogram Variance	Histogram variance	Independent	133	0.0 - 269.0	2126
Histogram Tendency	Histogram trend	Independent	3	0.0	2126
Fetal Health	Fetal health: 1 - Normal 2 - Suspect 3 - Pathological	Dependent	3	1.0 2.0 3.0	1655 295 176

Figure 4: Dataset Table

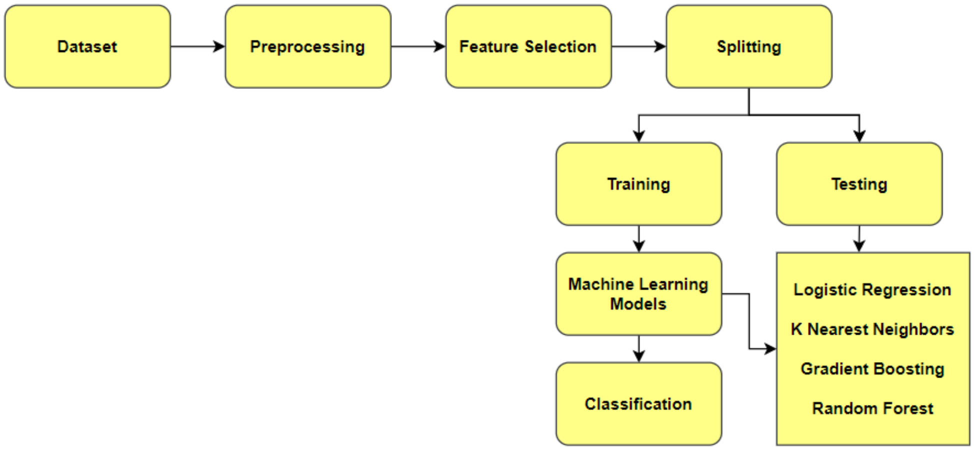

Figure 5: System Architectural Diagram

✓ DATA PREPARATION

In this step, in order to confirm the size of the dataset, we checked using Python Pandas data frame.

```
#Size of the dataset
print(f"The size of the dataset: {my_dataset.shape}")
```

```
The size of the dataset: (2126, 22)
```

Figure 6: Size of the Dataset

All the variables were of a numerical format which comes under the datatype float64. If any categorical variables are available in the dataset, it can be sorted using the methods ordinal encoding, one-hot encoding and label encoding.

```
#Datatypes

my_dataset.dtypes
```

```
baseline value                                            float64
accelerations                                             float64
fetal_movement                                            float64
uterine_contractions                                      float64
light_decelerations                                       float64
severe_decelerations                                      float64
prolongued_decelerations                                  float64
abnormal_short_term_variability                           float64
mean_value_of_short_term_variability                      float64
percentage_of_time_with_abnormal_long_term_variability    float64
mean_value_of_long_term_variability                       float64
histogram_width                                           float64
histogram_min                                             float64
histogram_max                                             float64
histogram_number_of_peaks                                 float64
histogram_number_of_zeroes                                float64
histogram_mode                                            float64
histogram_mean                                            float64
histogram_median                                          float64
histogram_variance                                        float64
histogram_tendency                                        float64
fetal_health                                              float64
dtype: object
```

Figure 7: Datatype

Then, we analyzed for the missing and null values in the dataset, the result seems to be zero. We use Missingno library which helps to visualize the distribution of NaN values. Missingno is a Python library and attuned with Pandas.

```
#Checking for missing and null values
missing_values = my_dataset.columns[my_dataset.isnull().any()]
print(f"Missing values are:\n{my_dataset[missing_values].isnull().sum()}")

null_values = my_dataset.columns[my_dataset.isna().any()]
print(f"Null values are:\n{my_dataset[null_values].isna().sum()}")
```

```
Missing values are:
Series([], dtype: float64)
Null values are:
Series([], dtype: float64)
```

Figure 8: Checking for Missing and Null Values

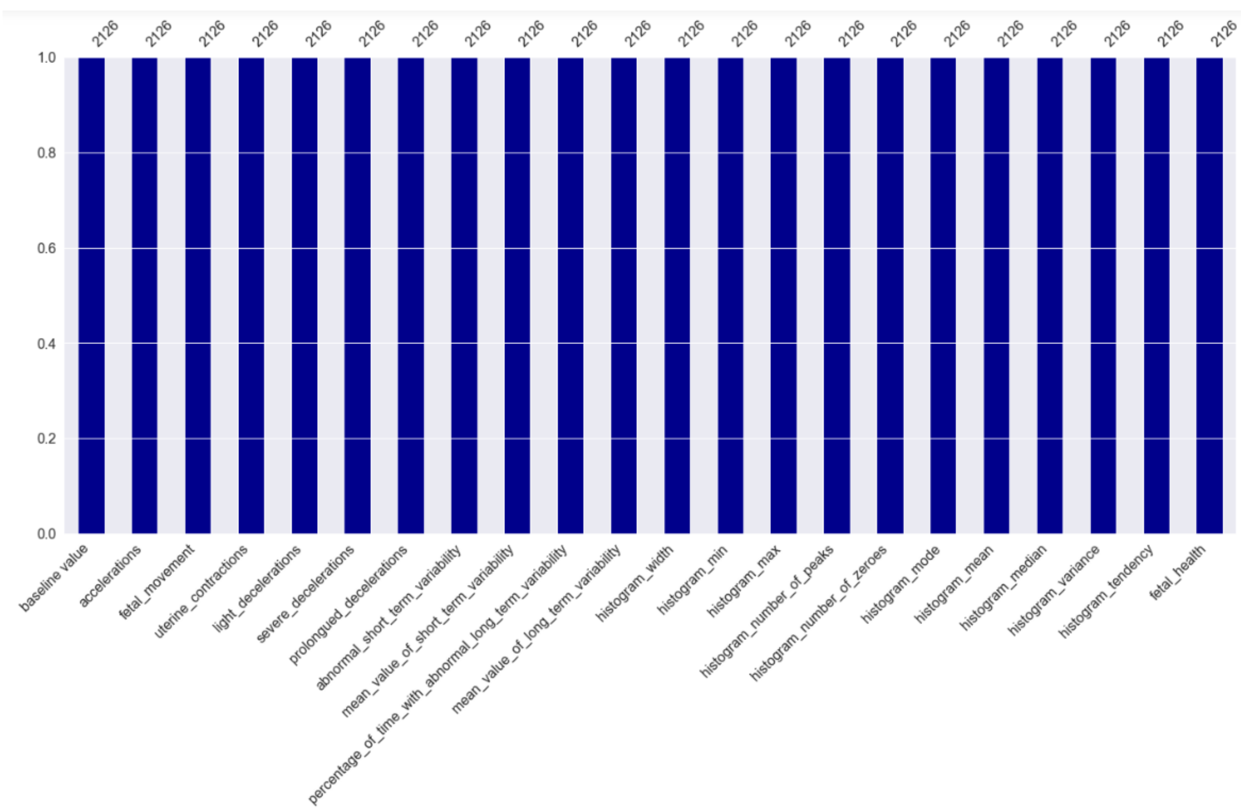

Figure 9: Null Count Analysis

✓ DATA EXPLORATION

The foremost step of exploration is describing the dataset by the method Univariate Analysis in Python using Matplotlib library, with that we can understand the in-depth calculation of the data count. The variance and range of the variables seems to be different, while the histograms indicate skewed data distribution than a Gaussian distribution.

```
#Univariate analysis
my_dataset.describe().T
```

	count	mean	std	min	25%	50%	75%	max
baseline value	2126.0	133.303857	9.840844	106.0	126.000	133.000	140.000	160.000
accelerations	2126.0	0.003178	0.003866	0.0	0.000	0.002	0.006	0.019
fetal_movement	2126.0	0.009481	0.046666	0.0	0.000	0.000	0.003	0.481
uterine_contractions	2126.0	0.004366	0.002946	0.0	0.002	0.004	0.007	0.015
light_decelerations	2126.0	0.001889	0.002960	0.0	0.000	0.000	0.003	0.015
severe_decelerations	2126.0	0.000003	0.000057	0.0	0.000	0.000	0.000	0.001
prolongued_decelerations	2126.0	0.000159	0.000590	0.0	0.000	0.000	0.000	0.005
abnormal_short_term_variability	2126.0	46.990122	17.192814	12.0	32.000	49.000	61.000	87.000
mean_value_of_short_term_variability	2126.0	1.332785	0.883241	0.2	0.700	1.200	1.700	7.000
percentage_of_time_with_abnormal_long_term_variability	2126.0	9.846660	18.396880	0.0	0.000	0.000	11.000	91.000
mean_value_of_long_term_variability	2126.0	8.187629	5.628247	0.0	4.600	7.400	10.800	50.700
histogram_width	2126.0	70.445908	38.955693	3.0	37.000	67.500	100.000	180.000
histogram_min	2126.0	93.579492	29.560212	50.0	67.000	93.000	120.000	159.000
histogram_max	2126.0	164.025400	17.944183	122.0	152.000	162.000	174.000	238.000
histogram_number_of_peaks	2126.0	4.068203	2.949386	0.0	2.000	3.000	6.000	18.000
histogram_number_of_zeroes	2126.0	0.323612	0.706059	0.0	0.000	0.000	0.000	10.000
histogram_mode	2126.0	137.452023	16.381289	60.0	129.000	139.000	148.000	187.000
histogram_mean	2126.0	134.610536	15.593596	73.0	125.000	136.000	145.000	182.000
histogram_median	2126.0	138.090310	14.466589	77.0	129.000	139.000	148.000	186.000
histogram_variance	2126.0	18.808090	28.977636	0.0	2.000	7.000	24.000	269.000
histogram_tendency	2126.0	0.320320	0.610829	-1.0	0.000	0.000	1.000	1.000
fetal_health	2126.0	1.304327	0.614377	1.0	1.000	1.000	1.000	3.000

Figure 10: Description of Dataset

```
#Evaluating distributions of the features
data_hist_plot = dataset.hist(figsize = (20,20), color = "#000088")
```

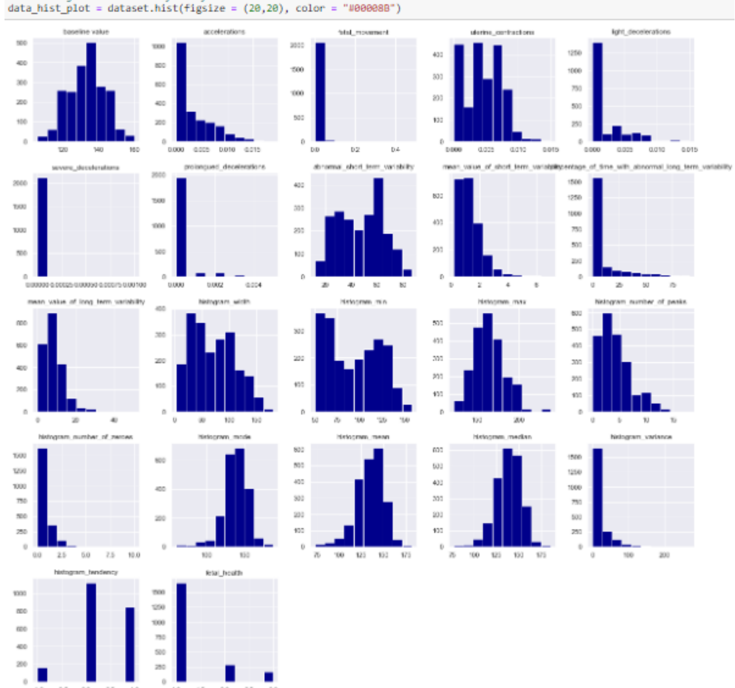

Figure 11: Evaluating Distributions of the Features

Added to, tried Kurtosis with dataset, which is nothing but a size of the joint mass of a distribution's ends comparative to the midpoint of the scattering. When a set of about normal data is graphed via a histogram, it shows an alarm top and most data within three standard deviations (plus or minus) of the mean.

```
#Kurtosis
kurtosis_value = my_dataset.kurt(axis=0)
print("Kurtosis:")
print(kurtosis_value)
```

```
Kurtosis:
baseline value                                            -0.292943
accelerations                                              0.767648
fetal_movement                                            64.260821
uterine_contractions                                      -0.635071
light_decelerations                                        2.517461
severe_decelerations                                     299.424142
prolongued_decelerations                                  20.515918
abnormal_short_term_variability                           -1.051030
mean_value_of_short_term_variability                       4.700756
percentage_of_time_with_abnormal_long_term_variability     4.252998
mean_value_of_long_term_variability                        4.131254
histogram_width                                           -0.902287
histogram_min                                             -1.290422
histogram_max                                              0.632769
histogram_number_of_peaks                                  0.504211
histogram_number_of_zeroes                                30.365084
histogram_mode                                             3.009531
histogram_mean                                             0.933427
histogram_median                                           0.667259
histogram_variance                                        15.131589
histogram_tendency                                        -0.652639
fetal_health                                               2.091215
dtype: float64
```

Figure 12: Kurtosis

The heat map shows features with light colors have high correlation with the target variable.

```
#Bivariate analysis

#Correlation Analysis of numeric features with output variable "Fetal health"
numeric_data = my_dataset.select_dtypes(exclude="object")
numeric_corr = numeric_data.corr()
f,ax=plt.subplots(figsize=(25,1))
sns.heatmap(numeric_corr.sort_values(by=["fetal_health"], ascending=False).head(1), cmap="RdBu")
plt.title("Numerical features correlation with the fetal_health", weight="bold", fontsize=18, color="#00008B")
plt.yticks(weight="bold", color="darkgreen", rotation=0)

plt.show()
```

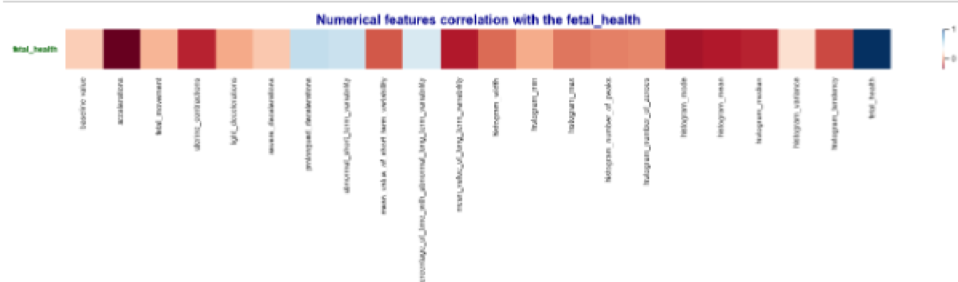

Figure 13: Correlation Analysis

The below code snippet proves that the prolongued decelerations, abnormal short term variability, percentage of time with abnormal long term variability are the features having high correlation with the target column (fetal health). The feature selection benefits to comprehend how the features are interconnected with each other.

```python
#Feature importance
Num_feature = numeric_corr["fetal_health"].sort_values(ascending=False).head(20).to_frame()

cm = sns.light_palette("#5F9EA0", as_cmap=True)

style = Num_feature.style.background_gradient(cmap=cm)
style
```

	fetal_health
fetal_health	1.000000
prolongued_decelerations	0.484859
abnormal_short_term_variability	0.471191
percentage_of_time_with_abnormal_long_term_variability	0.426146
histogram_variance	0.206630
baseline value	0.148151
severe_decelerations	0.131934
fetal_movement	0.088010
histogram_min	0.063175
light_decelerations	0.058870
histogram_number_of_zeroes	-0.016682
histogram_number_of_peaks	-0.023666
histogram_max	-0.045265
histogram_width	-0.068789
mean_value_of_short_term_variability	-0.103382
histogram_tendency	-0.131976
uterine_contractions	-0.204894
histogram_median	-0.205033
mean_value_of_long_term_variability	-0.226797
histogram_mean	-0.226985

Figure 14: Feature Importance

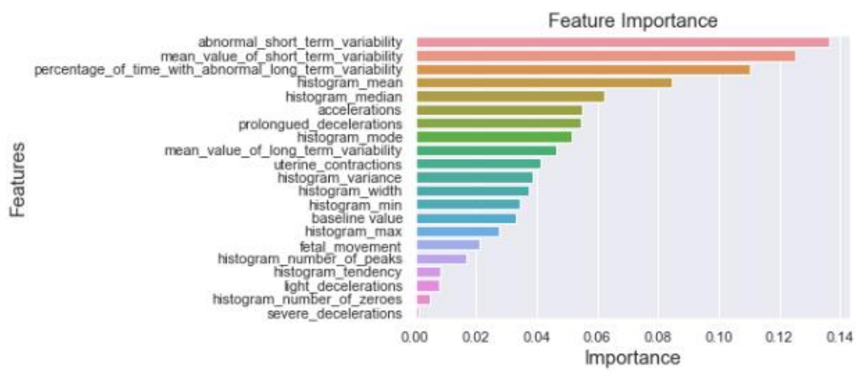

Figure 15: Graphical Representation of Feature Importance

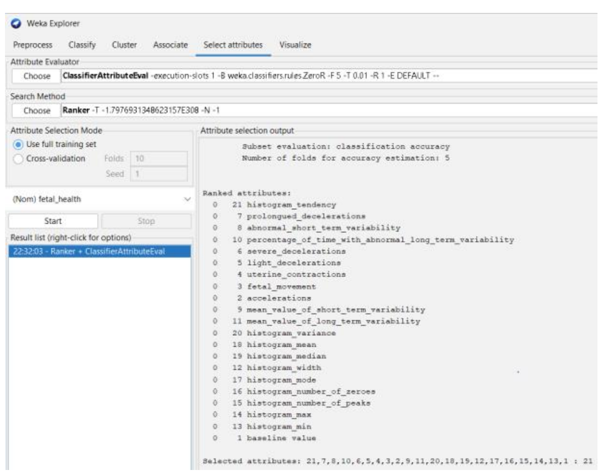

Figure 16: Attribute Evaluation in WEKA

Multivariate analysis has executed by Python in the form of heat map. A heat map is a two-dimensional depiction of statistics with the advantage of colors. Heat maps can help the handler imagine guileless or difficult figures. Correlation heat maps are perfect for linking the dimension for individual pair of dimension values. Also scatter plot matrix is performed which is a lattice (or matrix) of disseminate plots used to envisage bivariate interactions between amalgamations of variables.

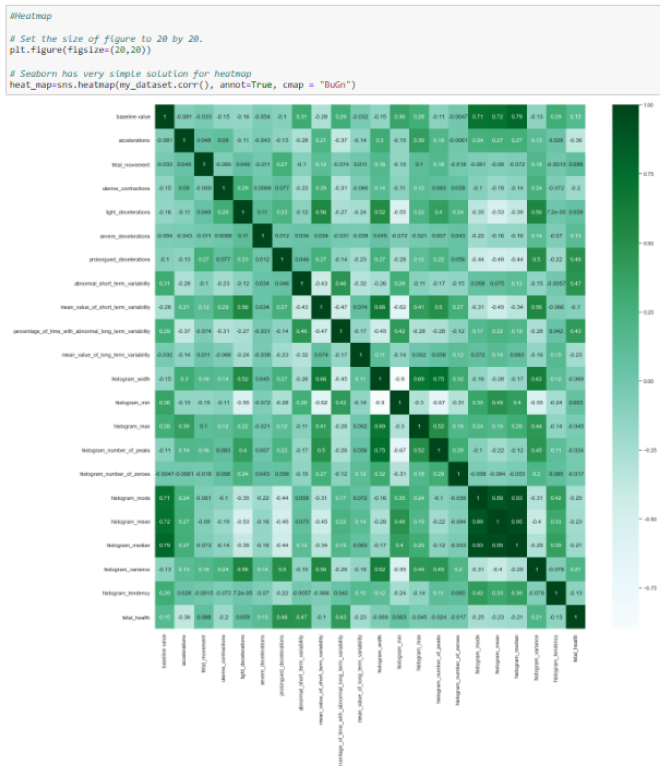

Figure 17: Heat Map

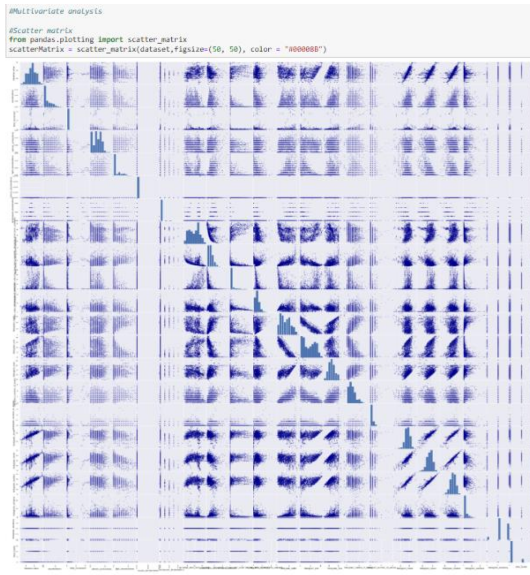

Figure 18: Scatter Matrix Plot

Feature scaling in machine learning is one of the greatest steps in the course of the pre-processing of information before creating a machine learning model. Scaling can sort a alteration amongst a frail machine learning model and a better one. The furthermost mutual modus operandi of feature scaling is Normalization and Standardization. Normalization is used once we want to certain our standards in the middle of two numbers, characteristically, between [0, 1] or [-1, 1]. Whereas Standardization transmutes the data to have nil mean and a variance of 1, they create our records unit less. In this case, I have used Standard Scaler from Sklearn Python library which is appropriate for non-gaussian distributions and different range of values.

```
#Feature scaling

#Setting up a standard scaler for the features and analyzing
headings = ['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']
scale_X = StandardScaler()
X =  pd.DataFrame(scale_X.fit_transform(my_dataset.drop(["fetal_health"],axis = 1),), columns = headings)
```

Figure 19: Feature Scaling with Standard Scaler

✓ DATA MODELLING AND VISUALIZATION

Data modelling is done with the 3 cases given in the target variable (1.0 – Normal, 2.0 – Suspect, 3.0 – Pathological). I have splatted the training data into 70% and testing data into 30%. The train test split methodology helps to capture the model performance much better. Also, cross-validation is a model validation modus operandi for evaluating how the results of a statistical exploration will oversimplify to an independent data set. To increase all scores for each ML model, we want to hunt the set of "hyper parameters" by using the common tactic "Grid search" for the selected four models. Classification methods were trained to predict the target cases of fetal health and their performance evaluated using the test data. The final model was carefully chosen by valuation of the metrics including the confusion matrix and recall.

```
# Importing train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 42, stratify = y)
X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

```
((1488, 21), (638, 21), (1488,), (638,))
```

Figure 20: Train Test Split

1) ***Logistic Regression***

This is a forecasting technique that makes use of independent variables to anticipate the value of a categorical dependent variable.

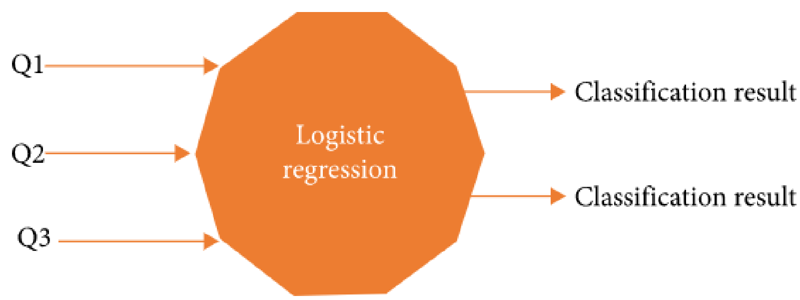

Figure 21: Logistic Regression Classifier

```
Classification Report
              precision    recall  f1-score   support

         1.0       0.93      0.95      0.94       497
         2.0       0.60      0.66      0.63        88
         3.0       0.95      0.70      0.80        53

    accuracy                           0.89       638
   macro avg       0.83      0.77      0.79       638
weighted avg       0.89      0.89      0.89       638
```

Figure 22: Classification Report by LR Model

```
Confusion Matrix:
[[470  26   1]
 [ 29  58   1]
 [  4  12  37]]
```

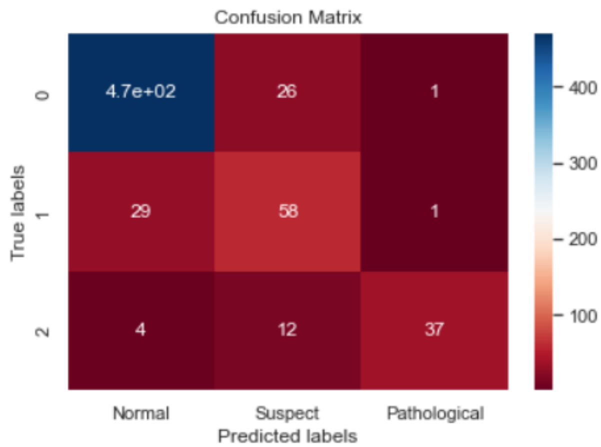

Figure 23: Confusion Matrix by LR Model

Since the target variable's 2.0 cases seems to be over fitted with low recall. Then tried the same model in WEKA, the recall for case 2.0 seems to be better.

```
=== Summary ===

Correctly Classified Instances        2124               99.9059 %
Incorrectly Classified Instances         2                0.0941 %
Kappa statistic                          0.9974
Mean absolute error                      0.0006
Root mean squared error                  0.0177
Relative absolute error                  0.2554 %
Root relative squared error              5.0567 %
Total Number of Instances             2126

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.004    0.999      1.000    0.999      0.997    1.000     1.000     1
                 0.993    0.000    1.000      0.993    0.997      0.996    1.000     1.000     2
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     3
Weighted Avg.    0.999    0.003    0.999      0.999    0.999      0.997    1.000     1.000

=== Confusion Matrix ===

    a    b    c   <-- classified as
 1655    0    0 |    a = 1
    2  293    0 |    b = 2
    0    0  176 |    c = 3
```

Figure 24: Logistic Regression Classifier in WEKA

2) K Nearest Neighbors

The K-NN algorithm keeps all available data and classifies new data points conditional on how equivalent they are to previous categorized data.

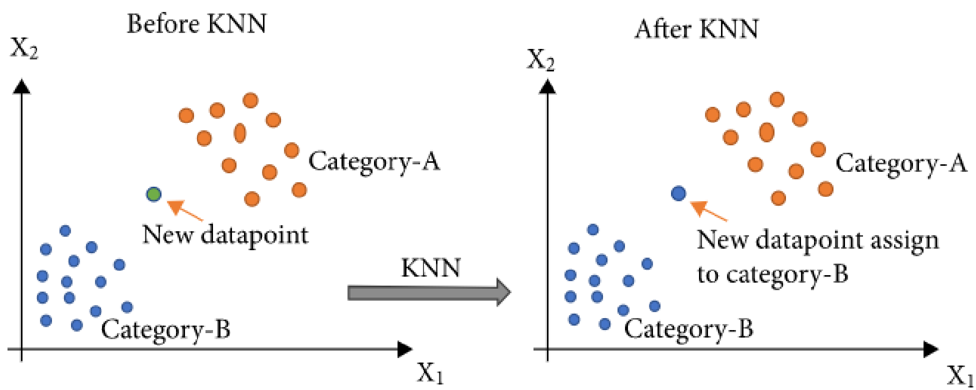

Figure 25: K Nearest Neighbors Classifier

Classification Report

	precision	recall	f1-score	support
1.0	0.94	0.96	0.95	497
2.0	0.66	0.66	0.66	88
3.0	0.88	0.68	0.77	53
accuracy			0.90	638
macro avg	0.83	0.77	0.79	638
weighted avg	0.90	0.90	0.89	638

Figure 26: Classification Report by KNN model

```
Confusion Matrix:
[[478  18   1]
 [ 26  58   4]
 [  5  12  36]]
```

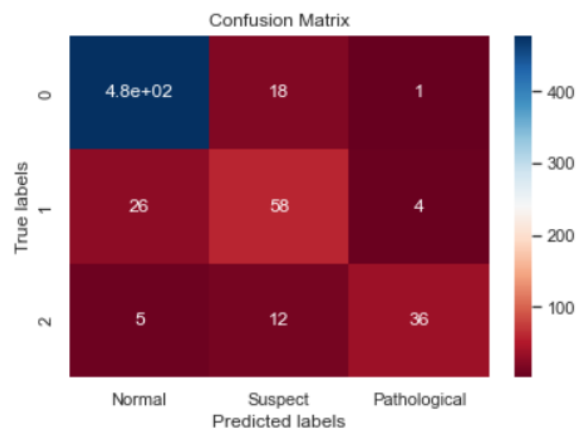

Figure 27: Confusion Matrix by KNN model

Same as LR model, the target variable's 2.0 cases seems to be over fitted with low recall. Then tried the same model in WEKA, the recall for case 2.0 seems to be better.

```
=== Summary ===

Correctly Classified Instances        2124               99.9059 %
Incorrectly Classified Instances         2                0.0941 %
Kappa statistic                          0.9974
Mean absolute error                      0.0012
Root mean squared error                  0.0177
Relative absolute error                  0.5085 %
Root relative squared error              5.0602 %
Total Number of Instances             2126

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.004    0.999      1.000    0.999      0.997    1.000     1.000     1
                 0.993    0.000    1.000      0.993    0.997      0.996    1.000     1.000     2
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     3
Weighted Avg.    0.999    0.003    0.999      0.999    0.999      0.997    1.000     1.000

=== Confusion Matrix ===

    a    b    c   <-- classified as
 1655    0    0 |    a = 1
    2  293    0 |    b = 2
    0    0  176 |    c = 3
```

Figure 28: KNN Classifier in WEKA

3) Gradient Boosting

Gradient boosting works by constructing weak prediction models consecutively where each model tries to predict the inaccuracy left over by the previous model.

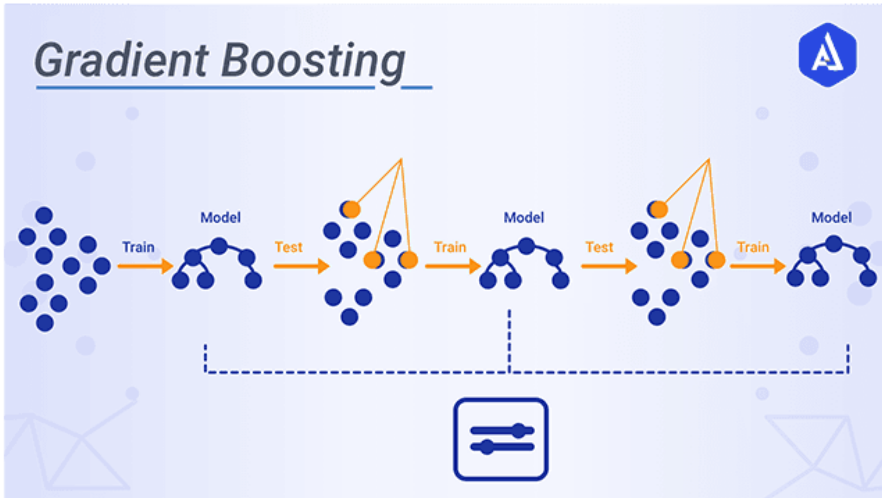

Figure 29: Gradient Boosting Model

Classification Report

	precision	recall	f1-score	support
1.0	0.95	0.96	0.95	497
2.0	0.69	0.69	0.69	88
3.0	0.78	0.68	0.73	53
accuracy			0.90	638
macro avg	0.80	0.78	0.79	638
weighted avg	0.90	0.90	0.90	638

Figure 30: Classification Report by GB model

```
Confusion Matrix:
[[476  15   6]
 [ 23  61   4]
 [  4  13  36]]
```

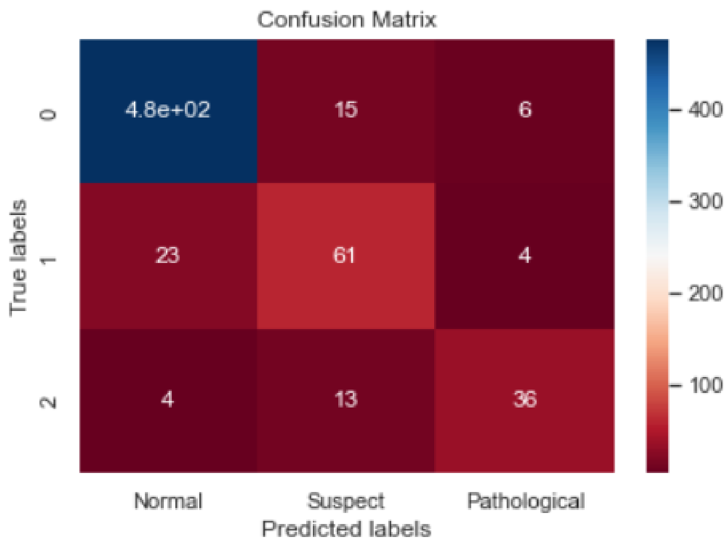

Figure 31: Confusion Matrix by GB Model

Same as previous 2 models, the target variable's 2.0 cases seems to be over fitted with low recall. Then tried the LMT model in WEKA, the recall for case 2.0 seems to be better.

```
=== Summary ===

Correctly Classified Instances        1874               88.1468 %
Incorrectly Classified Instances       252               11.8532 %
Kappa statistic                          0.6455
Mean absolute error                      0.1183
Root mean squared error                  0.2432
Relative absolute error                 48.166  %
Root relative squared error             69.4414 %
Total Number of Instances             2126

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.961    0.367    0.902      0.961    0.931      0.656    0.913     0.962     1
                 0.529    0.031    0.732      0.529    0.614      0.573    0.918     0.712     2
                 0.722    0.011    0.852      0.722    0.782      0.767    0.945     0.787     3
Weighted Avg.    0.881    0.291    0.874      0.881    0.874      0.654    0.916     0.913

=== Confusion Matrix ===

    a    b    c   <-- classified as
 1591   48   16 |    a = 1
  133  156    6 |    b = 2
   40    9  127 |    c = 3
```

Figure 32: LMT Classifier in WEKA

4) Random Forest

The best performance was achieved by random forest model, whereas the recall and accuracy seems to be good compared to other models. The random forest creates a large number of different trees and then combines them to provide a more accurate and reliable representation.

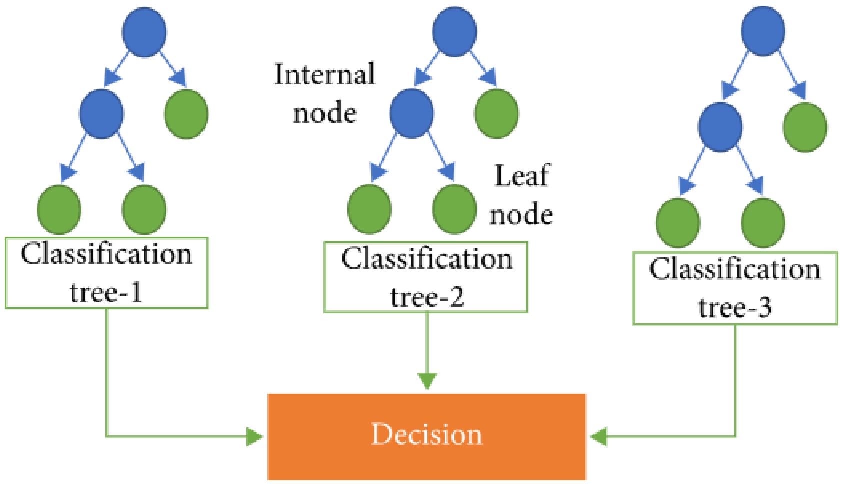

Figure 33: Random Forest Model

Classification Report

	precision	recall	f1-score	support
1.0	0.95	0.97	0.96	497
2.0	0.80	0.73	0.76	88
3.0	0.87	0.87	0.87	53
accuracy			0.93	638
macro avg	0.87	0.86	0.86	638
weighted avg	0.93	0.93	0.93	638

Figure 34: Classification Report by RF Model

```
Confusion Matrix:
[[482  11   4]
 [ 21  64   3]
 [  2   5  46]]
```

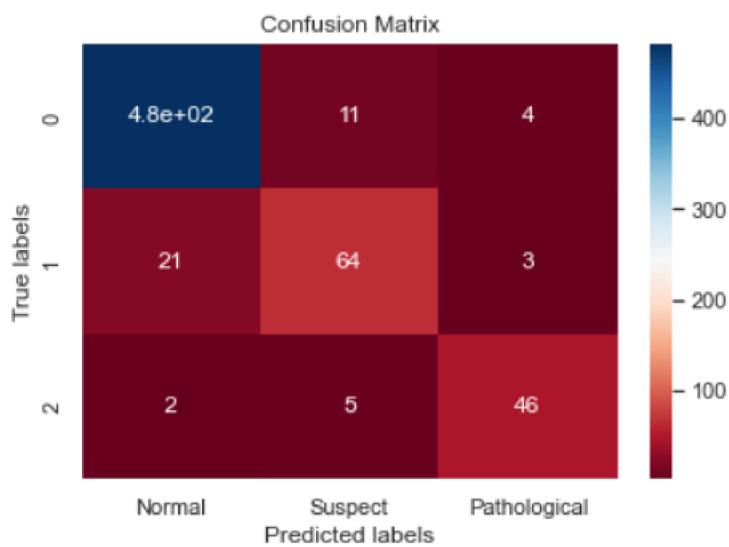

Figure 35: Confusion Matrix by RF Model

Also, tried the same model in WEKA, the recall and accuracy seems to be good.

```
=== Summary ===

Correctly Classified Instances        2124               99.9059 %
Incorrectly Classified Instances         2                0.0941 %
Kappa statistic                          0.9974
Mean absolute error                      0.0569
Root mean squared error                  0.0935
Relative absolute error                 23.1535 %
Root relative squared error             26.6969 %
Total Number of Instances             2126

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.004    0.999      1.000    0.999      0.997    1.000     1.000     1
                 0.993    0.000    1.000      0.993    0.997      0.996    1.000     1.000     2
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     3
Weighted Avg.    0.999    0.003    0.999      0.999    0.999      0.997    1.000     1.000

=== Confusion Matrix ===

    a    b    c   <-- classified as
 1655    0    0 |    a = 1
    2  293    0 |    b = 2
    0    0  176 |    c = 3
```

Figure 36: Random Forest Classifier in WEKA

✓ Model Evaluation

The below table clearly shows that among the many models in the framework, Random Forest is the best. It has a higher F1-score and has greater exactness, review, and the region beneath the bend. The gradient boosting also achieved 99% accuracy in this paper, but in low recall in case 2.0 of the target variable. Logistic Regression accomplished the lowest accuracy of 90 percent.

Score	Model
0.999328	Random Forest
0.997984	Gradient Boosting Classifier
0.956317	KNN
0.906586	Logistic Regression

Figure 37: Model Comparison

Conclusion

The correctness measurement of the models used in this research is much superior to that of earlier examinations, symptomatic of that the models used in this investigation are more upright. Abundant model assessments have shown their vigor, and the system may be inferred from the exploration scrutiny. In the future, altered intricate machine learning models can be applied to make this system more strong.

References

1) Alam, M.T., Khan, M.A.I., Dola, N.N., Tazin, T., Khan, M.M., Albraikan, A.A. and Almalki, F.A. (2022). Comparative Analysis of Different Efficient Machine Learning Methods for Fetal Health Classification. *Applied Bionics and Biomechanics*, 2022, pp.1–12. doi:10.1155/2022/6321884.
2) Buitendijk, S., Zeitlin, J., Cuttini, M., Langhoff-Roos, J. and Bottu, J. (2003). Indicators of fetal and infant health outcomes. *European Journal of Obstetrics & Gynecology and Reproductive Biology*, [online] 111, pp.S66–S77. doi:10.1016/j.ejogrb.2003.09.007.
3) elayer (2022a). *Fetal-Health-Classifier-Project/Fetal Health Classification - EDA .ipynb at main · elayer/Fetal-Health-Classifier-Project*. [online] GitHub. Available at: https://github.com/elayer/Fetal-Health-Classifier-Project/blob/main/Fetal%20Health%20Classification%20-%20EDA%20.ipynb [Accessed 1 Sep. 2022].

4) elayer (2022b). *Fetal-Health-Classifier-Project/Fetal Health Classification - Model Building - v2.ipynb at main · elayer/Fetal-Health-Classifier-Project*. [online] GitHub. Available at: https://github.com/elayer/Fetal-Health-Classifier-Project/blob/main/Fetal%20Health%20Classification%20-%20Model%20Building%20-%20v2.ipynb [Accessed 1 Sep. 2022].
5) Ghosh, R. (2021). *Prediction of Fetal Health Classification using Machine Learning*. [online] Medium. Available at: https://ghoshratul063.medium.com/prediction-of-fetal-health-classification-using-machine-learning-6dac54521e3c [Accessed 1 Sep. 2022].
6) Hasan, B., Hoodbhoy, Z., Noman, M., Shafique, A., Nasim, A. and Chowdhury, D. (2019). Use of machine learning algorithms for prediction of fetal risk using cardiotocographic data. *International Journal of Applied and Basic Medical Research*, 9(4), p.226. doi:10.4103/ijabmr.ijabmr_370_18.
7) Jose, J.K. (2021). *Fetal-Health-Status-Prediction-Using-multiple-Machine-Learning-algorithms-using-Python*. [online] GitHub. Available at: https://github.com/johnykjose/Fetal-Health-Status-Prediction-Using-multiple-Machine-Learning-algorithms-using-Python/blob/main/fetal-health-status-classification-99-accuracy.ipynb [Accessed 1 Sep. 2022].
8) kaggle.com. (n.d.). *Fetal Health - EDA and Prediction*. [online] Available at: https://www.kaggle.com/code/misaelcribeiro/fetal-health-eda-and-prediction.
9) kaggle.com. (n.d.). *Fetal Health Classification Prediction*. [online] Available at: https://www.kaggle.com/code/khalidhussain/fetal-health-classification-prediction [Accessed 1 Sep. 2022].
10) kaggle.com. (n.d.). *Fetal Health classification-A comparison using PCA*. [online] Available at: https://www.kaggle.com/code/meeraajayakumar/fetal-health-classification-a-comparison-using-pca [Accessed 1 Sep. 2022].
11) kaggle.com. (n.d.). *Fetal Health Prediction — 94% accuracy*. [online] Available at: https://www.kaggle.com/code/luishpinto/fetal-health-prediction-94-accuracy [Accessed 1 Sep. 2022].
12) kaggle.com. (n.d.). *FetalHealthClassification 99%*. [online] Available at: https://www.kaggle.com/code/casper6290/fetalhealthclassification-99 [Accessed 1 Sep. 2022].
13) kaggle.com. (n.d.). *FHC-EDA and Supervised Learning Algorithms*. [online] Available at: https://www.kaggle.com/code/naubergois/fhc-eda-and-supervised-learning-algorithms/data [Accessed 1 Sep. 2022].
14) kaggle.com. (n.d.). *Random Forest: Fetal Health Classification*. [online] Available at: https://www.kaggle.com/code/radhikapotey/random-forest-fetal-health-classification/data [Accessed 1 Sep. 2022].

15) kaggle.com. (n.d.). *Step by Step Fetal Health Prediction-99%-Detailed*. [online] Available at: https://www.kaggle.com/code/pariaagharabi/step-by-step-fetal-health-prediction-99-detailed [Accessed 1 Sep. 2022].
16) Layer, E. (2022). *elayer/Fetal-Health-Classifier-Project*. [online] GitHub. Available at: https://github.com/elayer/Fetal-Health-Classifier-Project [Accessed 1 Sep. 2022].
17) Li, J. and Liu, X. (2021). *Fetal Health Classification Based on Machine Learning*. [online] IEEE Xplore. doi:10.1109/ICBAIE52039.2021.9389902.
18) Modernanalyst.com. (2015). *Describe the difference between univariate, bivariate and multivariate analysis?* [online] Available at: https://www.modernanalyst.com/Careers/InterviewQuestions/tabid/128/ID/4904/Describe-the-difference-between-univariate-bivariate-and-multivariate-analysis.aspx.
19) Naveen (2019). *What Is Data Analytics?* [online] Intellipaat Blog. Available at: https://intellipaat.com/blog/what-is-data-analytics/.
20) Navuluri, N.R. (2021). Fetal Health Prediction using Classification Techniques. *International Journal of Engineering Research & Technology*, [online] 10(11). doi:10.17577/IJERTV10IS110182.
21) Peterek, T., Gajdoš, P., Dohnálek, P. and Krohová, J. (2014). Human Fetus Health Classification on Cardiotocographic Data Using Random Forests. *Advances in Intelligent Systems and Computing*, pp.189–198. doi:10.1007/978-3-319-07773-4_19.
22) Simran (2022). *Fetal Arc: Predicting Fetal Health, and Birth-Weight of the fetus using Machine Learning*. [online] Analytics Vidhya. Available at: https://medium.com/analytics-vidhya/fetal-arc-predicting-fetal-health-and-birth-weight-of-the-fetus-using-machine-learning-5d6726323904 [Accessed 1 Sep. 2022].
23) Systems, A.T. & (2020). *Fetal health classification — on cAInvas*. [online] AITS Journal. Available at: https://medium.com/ai-techsystems/fetal-health-classification-on-cainvas-1129886daa19 [Accessed 1 Sep. 2022].
24) www.akira.ai. (n.d.). *Introduction to Gradient Boosting Machines - Akira AI*. [online] Available at: http://www.akira.ai/glossary/gradient-boosting [Accessed 1 Sep. 2022].
25) www.digitalocean.com. (n.d.). *EDA - Exploratory Data Analysis: Using Python Functions | DigitalOcean*. [online] Available at: https://www.digitalocean.com/community/tutorials/exploratory-data-analysis-python.
26) www.kaggle.com. (n.d.). *Fetal Health Classification*. [online] Available at: https://www.kaggle.com/datasets/andrewmvd/fetal-health-classification.
27) www.scikit-yb.org. (n.d.). *Confusion Matrix — Yellowbrick v1.5 documentation*. [online] Available at: https://www.scikit-yb.org/en/latest/api/classifier/confusion_matrix.html [Accessed 1 Sep. 2022].