JAWAHARLAL NEHRU UNIVERSITY

School of Computer and System Sciences

Session 2020-2021

Semester II

Wireless Sensor Networks [CS-770] Project

on

Fetal Health Classification

By-

Gargi Mishra

20/10/MT/017

M.Tech Computer Science and Technology

Submitted to-

Dr. Poonam Agarwal

Table of Contents

1. INTRODUCTION

Having a healthy pregnancy is one of the best ways to promote a healthy birth. Getting early and regular prenatal care improves the chances of a healthy pregnancy. This care can begin even before pregnancy with a pre-pregnancy care.

A normal pregnancy lasts nine months. Each three-month period of pregnancy is called a trimester. During each trimester, the fetus grows and develops. Regular medical checkups and prenatal tests are very important. They can [1]

- Help keep other and baby healthy
- Spot problems with baby (if there are any). In some cases, health care professionals can treat the problem before baby is born. But even when they cannot, it can still be helpful to know about the problem early on. That gives time to learn about baby's condition and prepare for any challenges one may face after the baby is born.
- Prevent problems during delivery. For example, if baby is breech (bottom first or feet first, instead of head first), one may need to have a Cesarean section to avoid complications.

There are two methods of fetal heart rate monitoring in labor. Auscultation is a method of periodically listening to the fetal heartbeat. Electronic fetal monitoring is a procedure in which instruments are used to continuously record the heartbeat of the fetus and the contractions of the woman's uterus during labor. The method that is used depends on the policy of ob-gyn or hospital, risk of problems, and how labor is going. If one do not have any complications or risk factors for problems during labor, either method is acceptable. [2]

2. LITERATURE REVIEW

2.1. History of Fetal Monitoring

It is thought that the fetal heartbeat was first heard in the middle of the 17th or 18th century by placing an ear to the mother's abdomen. However, in 1822, fetal heart rate monitoring during labor became generally accepted with Lejumeau de Kergaradec's use of the stethoscope. The first fetal electrocardiogram (EKG) recording took place in 1906. In 1958, Dr. Hon from Yale University first identified fetal distress by monitoring the fetal heart rate continuously through the mother's abdomen.[3]

2.2. Electronic Fetal Monitor (EFM)

Electronic fetal monitoring (EFM), also called ***cardiotocography*** **(CTG)**, is when the baby's heart rate is monitored with an ultrasound machine while the mother's contractions are monitored with a pressure sensor. Both of these sensors are linked to a recording machine, which shows a print-out or computer screen of the baby's heart rate and the mother's contractions shown together, often called *EFM tracings*. The monitor is assessing the baseline fetal heart rate and how it changes with contractions. It records any increases in the fetal heart rate (accelerations) and any decreases (decelerations), as well as the frequency and duration of the mother's uterine contractions.[3]

Methods of CTG

External cardiotocography can be used for continuous or intermittent monitoring. The fetal heart rate and the activity of the uterine muscle are detected by two transducers placed on the mother's abdomen, with one above the fetal heart to monitor heart rate, and the other at the fundus of the uterus to measure frequency of contractions. Doppler ultrasound provides the information, which is recorded on a paper strip known as a cardiotocograph (CTG). External tocometry is useful for showing the beginning and end of contractions as well as their frequency, but not the strength of the contractions. The absolute values of pressure readings on an external tocometer are dependent on position and are not sensitive in people who are obese.[4]

Internal cardiotocography uses an electronic transducer connected directly to the fetus. A wire electrode, sometimes called a spiral or scalp electrode, is attached to the fetal scalp through the cervical opening and is connected to the monitor. Internal monitoring provides a more accurate and consistent transmission of the fetal heart rate, as unlike external monitoring, it is not affected by factors such as movement. Internal monitoring may be used when external monitoring is inadequate, or if closer surveillance is needed. Internal tocometry can only be used if the amniotic sac is ruptured (either spontaneously or artificially) and the cervix is open. To gauge the strength of contractions, a small catheter (called an intrauterine pressure catheter or IUPC) is passed into the uterus past the fetus. Combined with an internal fetal monitor, an IUPC may give a more precise reading of the baby's heart rate and the strength of contractions. [5]

A typical CTG reading is printed on paper and may be stored on a computer for later reference. A variety of systems for centralized viewing of CTG have been installed in maternity hospitals in industrialized countries, allowing simultaneous monitoring of multiple tracings in one or more locations. Display of maternal vital signs, ST signals and an electronic partogram are available in the majority of these systems. A few of them have incorporated computer analysis of cardiotocographic signals or combined cardiotocographic and ST data analysis. [6]

2.3. Involvement of Machine Learning

When it comes to developing a machine learning model to derive some useful information, we first need to study the type of data being used and the factors which may affect the results. Is our case the data is collected from various CTG and our task is to predict whether a fatal is in normal, suspected or pathologic state. As it is obvious that the number of normal cases will be highest and the number of pathologic cases will be rare, we can say that the nature of data is imbalanced and we need to handle the skewness. There are many algorithmic approaches to handle this, some of them are discussed here.

To deal with the skewness, a balancing approach is required which will make the distribution in a ratio of 1:1 between the two classes. This way the difference between majority and minority classes can be recognizes easily which eventually make the classification easy. The balancing approach can mainly be classified into two categories, which can be further be classified into sub categories. Figure below shows this classification.

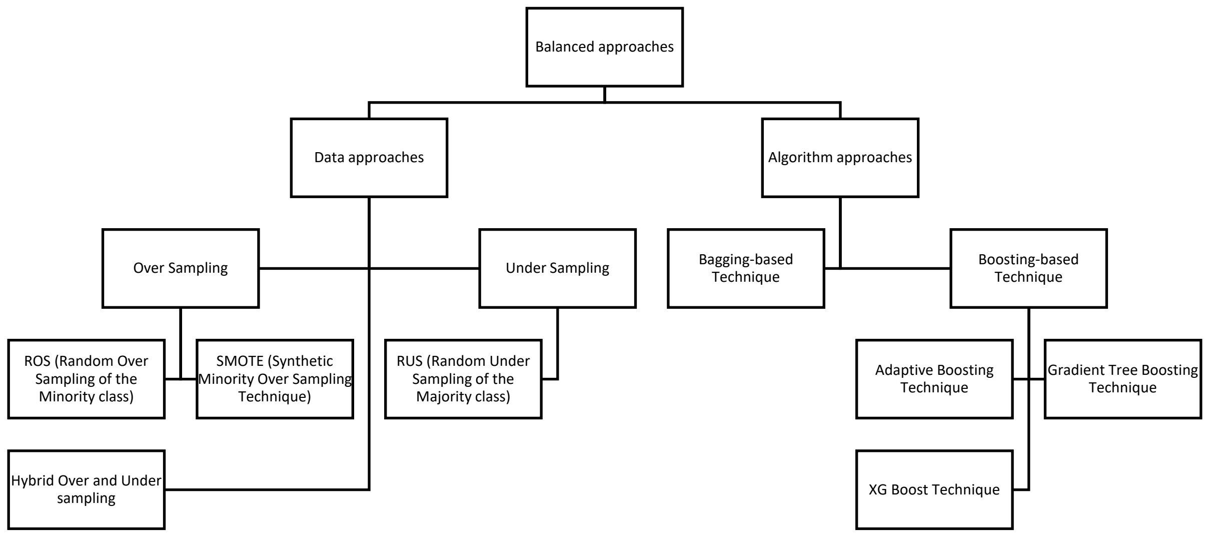


Figure 1: Imbalanced class data balancing approaches

In our project, we have used algorithmic level methods

Algorithm level methods

This method deals with modifying the already existing classification techniques to make them suitable to deal with imbalanced dataset. The main aim of this method is to improve the performance of classifiers and then construct a combined classifier which is based on the original data. The combination collectively predicts the final result.

- In **bagging-based techniques**, algorithm involve the generation of 'n' different training samples with replacement. The model is then trained on each algorithm separately and then aggregating the results.
- **Boosting** involves the concept of combining weak learners together to make a strong learner. Process starts with a base(weak) classifier whose prediction accuracy is considered as an average. In the next iteration the focus is on the results which were predicted incorrectly in the previous iteration. Same repeats for other iterations as well, this way we improve accuracy of results with each step.

3. PROJECT IMPLEMENTATION

3.1. Dataset

The dataset consists of measurements of fetal heart rate (FHR) and uterine contraction (UC) features on cardiotocograms classified by expert obstetricians. [7]

Dataset Info: 2126 fetal cardiotocograms (CTGs) were automatically processed and the respective diagnostic features measured. The CTGs were also classified by three expert obstetricians and a consensus classification label assigned to each of them. Classification was both with respect to a morphologic pattern (A, B, C. ...) and to a fetal state (N, S, P). Therefore the dataset can be used either for 10-class or 3-class experiments. [7]

Attributes of dataset

- LB - FHR baseline (beats per minute)
- AC - # of accelerations per second
- FM - # of fetal movements per second
- UC - # of uterine contractions per second
- DL - # of light decelerations per second
- DS - # of severe decelerations per second
- DP - # of prolongued decelerations per second
- ASTV - percentage of time with abnormal short term variability
- MSTV - mean value of short term variability
- ALTV - percentage of time with abnormal long term variability
- MLTV - mean value of long term variability
- Width - width of FHR histogram
- Min - minimum of FHR histogram
- Max - Maximum of FHR histogram
- Nmax - # of histogram peaks
- Nzeros - # of histogram zeros
- Mode - histogram mode

- Mean - histogram mean
- Median - histogram median
- Variance - histogram variance
- Tendency - histogram tendency
- CLASS - FHR pattern class code (1 to 10)
- NSP - fetal state class code (N=normal; S=suspect; P=pathologic)

3.2. Correlation among features

"Correlation" is a statistical term describing the degree to which two variables move in coordination with one-another.

Correlation coefficient: These are indicators of the strength of the linear relationship between two different variables. Its values range between -1.0 and 1.0.

Here we have used the correlation of all the features with target class to find the most relevant features which can be used in our model building.

3.3. Machine learning models implemented

We split the dataset into train data and test data in 80:20 proportion. Used the train data in model training and used test data to predict results.

In this project we have developed following models:

- Logistic Regression
- Support Vector Machine
- Decision Tree Classifier
- Random Forest Classifier
- Gradient Booster Classifier

4. RESULTS

To evaluate our model, we have used accuracy score as the parameter to evaluate our model, the results are given below:

```
Classification model with correlation:

Logistic Regression  0.9906103286384976
Support Vector Machine  0.9953051643192489
Decision Tree Classifier  1.0
Random Forest Classifier  0.9953051643192489
Gradient Booster Classifier  1.0
```

5. CONCLUSION

As we see from results, Gradient Booster Classifier performs best among all the models. Hence, we can say that boosting method of ensemble gives best result.

For classification of fetal health, a boosting method like Gradient Booster is recommended.

6. SOURCE CODE

WSN project - Fetal Health Classification

By- Gargi Mishra

Enroll no. - 20/10/MT/017

M.Tech Computer Science and Technology

SCSS, JNU

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
df = pd.read_csv('fetal_health.csv')
print(df.shape)
df.head()
```

```
(2126, 22)
```

Out[2]:

	baseline value	accelerations	fetal_movement	uterine_contractions	light_decelerations	severe_decelerations	prolongu
0	120.0	0.000	0.0	0.000	0.000	0.0	
1	132.0	0.006	0.0	0.006	0.003	0.0	
2	133.0	0.003	0.0	0.008	0.003	0.0	
3	134.0	0.003	0.0	0.008	0.003	0.0	
4	132.0	0.007	0.0	0.008	0.000	0.0	

5 rows × 22 columns

Data preprocessing

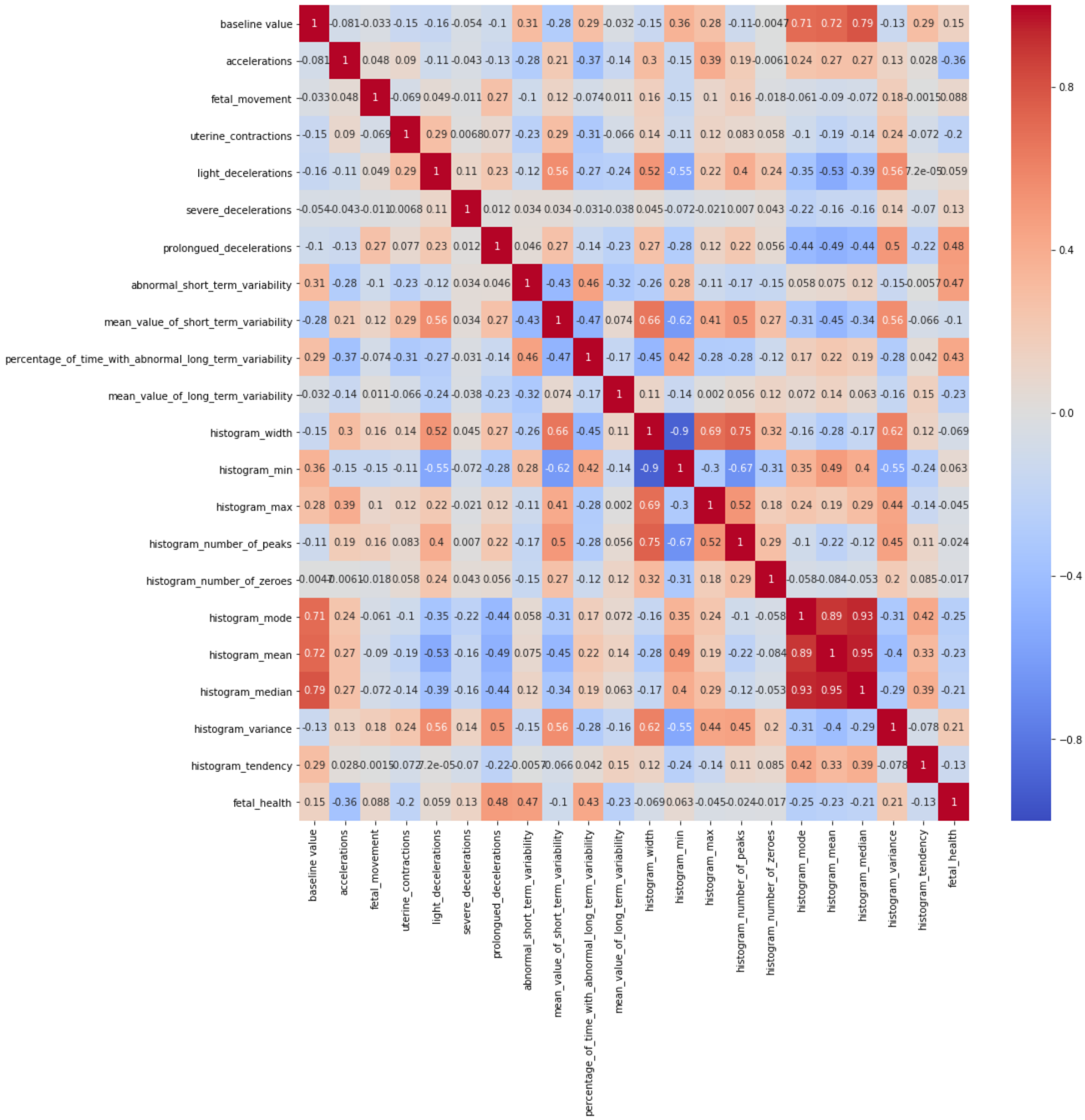
In [3]:
```
#missing values

df.isnull().sum()
```

Out[3]:
```
baseline value                                            0
accelerations                                             0
fetal_movement                                            0
uterine_contractions                                      0
light_decelerations                                       0
severe_decelerations                                      0
prolongued_decelerations                                  0
abnormal_short_term_variability                           0
mean_value_of_short_term_variability                      0
percentage_of_time_with_abnormal_long_term_variability    0
mean_value_of_long_term_variability                       0
histogram_width                                           0
histogram_min                                             0
histogram_max                                             0
histogram_number_of_peaks                                 0
histogram_number_of_zeroes                                0
histogram_mode                                            0
histogram_mean                                            0
histogram_median                                          0
histogram_variance                                        0
histogram_tendency                                        0
fetal_health                                              0
dtype: int64
```

In [4]:
```
#Correlation of features

corr = df.corr()
ax, fig = plt.subplots(figsize=(15,15))
sns.heatmap(corr, vmin=-1, cmap='coolwarm', annot=True)
plt.show()
```



In [5]:
```
#Correlation of target class 'fetal_health' with all the features

corr['fetal_health']
```

Out[5]:
```
baseline value                                            0.148151
accelerations                                            -0.364066
fetal_movement                                            0.088010
uterine_contractions                                     -0.204894
light_decelerations                                       0.058870
severe_decelerations                                      0.131934
prolongued_decelerations                                  0.484859
abnormal_short_term_variability                           0.471191
mean_value_of_short_term_variability                     -0.103382
percentage_of_time_with_abnormal_long_term_variability    0.426146
mean_value_of_long_term_variability                      -0.226797
histogram_width                                          -0.068789
histogram_min                                             0.063175
histogram_max                                            -0.045265
histogram_number_of_peaks                                -0.023666
histogram_number_of_zeroes                               -0.016682
histogram_mode                                           -0.250412
histogram_mean                                           -0.226985
histogram_median                                         -0.205033
histogram_variance                                        0.206630
histogram_tendency                                       -0.131976
fetal_health                                              1.000000
Name: fetal_health, dtype: float64
```

In [6]:
```
#Features having correlation > 10% with target class

corr[abs(corr['fetal_health']) > 0.1]['fetal_health']
```

Out[6]:
```
baseline value                                            0.148151
accelerations                                            -0.364066
uterine_contractions                                     -0.204894
severe_decelerations                                      0.131934
prolongued_decelerations                                  0.484859
abnormal_short_term_variability                           0.471191
mean_value_of_short_term_variability                     -0.103382
percentage_of_time_with_abnormal_long_term_variability    0.426146
mean_value_of_long_term_variability                      -0.226797
histogram_mode                                           -0.250412
histogram_mean                                           -0.226985
histogram_median                                         -0.205033
histogram_variance                                        0.206630
histogram_tendency                                       -0.131976
fetal_health                                              1.000000
Name: fetal_health, dtype: float64
```

In [7]:
```
x_corr = df[corr[abs(corr['fetal_health']) > 0.1]['fetal_health'].index]
x_corr = (x_corr-x_corr.mean())/x_corr.std()
y = df['fetal_health']
```

Classification models

In [8]:
```
#Spliting data into training and testing data
from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test=train_test_split(x_corr,y,random_state=1,test_size=0.2)
```

Logistic Regression

In [9]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

lr=LogisticRegression(max_iter=10000)
lr.fit(x_train,y_train)
y_lr=lr.predict(x_test)
acc_lr_c=accuracy_score(y_test,y_lr)
print("Linear Regression Success Rate :", "{:.2f}%".format(100*acc_lr_c))
```

```
Linear Regression Success Rate : 99.06%

C:\Anaconda\lib\site-packages\sklearn\linear_model\logistic.py:433: FutureWarning: Default s
olver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
C:\Anaconda\lib\site-packages\sklearn\linear_model\logistic.py:460: FutureWarning: Default m
ulti_class will be changed to 'auto' in 0.22. Specify the multi_class option to silence this
warning.
  "this warning.", FutureWarning)
```

Support Vector Machine

In [10]:
```
from sklearn.svm import SVC

svm=SVC()
svm.fit(x_train,y_train)
y_svm=svm.predict(x_test)
acc_svm_c=accuracy_score(y_test,y_svm)
print("Support Vector Machine Success Rate :", "{:.2f}%".format(100*acc_svm_c))
```

```
Support Vector Machine Success Rate : 99.53%

C:\Anaconda\lib\site-packages\sklearn\svm\base.py:196: FutureWarning: The default value of g
amma will change from 'auto' to 'scale' in version 0.22 to account better for unscaled featu
res. Set gamma explicitly to 'auto' or 'scale' to avoid this warning.
  "avoid this warning.", FutureWarning)
```

Decision Tree Classifier

In [11]:
```
from sklearn.tree import DecisionTreeClassifier

list1 = []
for leaves in range(2,10):
    classifier = DecisionTreeClassifier(max_leaf_nodes = leaves, random_state=0, criterion='e
ntropy')
    classifier.fit(x_train, y_train)
    y_dt = classifier.predict(x_test)
    list1.append(accuracy_score(y_test,y_dt)*100)
acc_dt_c=max(list1)
print("Top Decision Tree Classifier Success Rate:", "{:.2f}%".format(acc_dt_c))
```

```
Top Decision Tree Classifier Success Rate: 100.00%
```

Random Forest Classifier

In [12]:
```
from sklearn.ensemble import RandomForestClassifier

rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
y_rfc=rfc.predict(x_test)
acc_rfc_c=accuracy_score(y_test,y_rfc)
print("Random Forest Classifier Success Rate :", "{:.2f}%".format(100*acc_rfc_c))
```

```
Random Forest Classifier Success Rate : 99.53%

C:\Anaconda\lib\site-packages\sklearn\ensemble\forest.py:246: FutureWarning: The default val
ue of n_estimators will change from 10 in version 0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
```

Gradient Booster Classifier

In [13]:
```
from sklearn.ensemble import GradientBoostingClassifier

gbc=GradientBoostingClassifier()
gbc.fit(x_train,y_train)
y_gbc=gbc.predict(x_test)
acc_gbc_c=accuracy_score(y_test,y_gbc)
print("Gradient Booster Classifier Success Rate :", "{:.2f}%".format(100*acc_gbc_c))
```

```
Gradient Booster Classifier Success Rate : 100.00%
```

Results

In [14]:
```
print("Classification model with correlation:\n")
print("Logistic Regression ", acc_lr_c,"\nSupport Vector Machine ", acc_svm_c, "\nDecision Tr
ee Classifier ",acc_dt_c/100, "\nRandom Forest Classifier ",acc_rfc_c, "\nGradient Booster Cl
assifier ", acc_gbc_c)
```

```
Classification model with correlation:

Logistic Regression  0.9906103286384976
Support Vector Machine  0.9953051643192489
Decision Tree Classifier  1.0
Random Forest Classifier  0.9953051643192489
Gradient Booster Classifier  1.0
```

7. REFERENCES

[1] https://medlineplus.gov/fetalhealthanddevelopment.html

[2] https://www.acog.org/womens-health/faqs/fetal-heart-rate-monitoring-during-labor#:~:text=Auscultation%20is%20a%20method%20of,the%20woman's%20uterus%20during%20labor

[3] https://evidencebasedbirth.com/fetal-monitoring/

[4] Alfirevic, Zarko; Devane, Declan; Gyte, Gillian M. L.; Cuthbert, Anna (3 February 2017). "Continuous cardiotocography (CTG) as a form of electronic fetal monitoring (EFM) for fetal assessment during labour". *Cochrane Database of Systematic Reviews*.

[5] "Types of Fetal Heart Monitoring". *www.hopkinsmedicine.org*. Retrieved 21 March 2018.

[6] https://en.wikipedia.org/wiki/Cardiotocography

[7] https://archive.ics.uci.edu/ml/datasets/cardiotocography