Fetal Health Prediction

USING DECISION TREES, NEURAL NETWORKS, SVM

Nikhil Rongala | CS6375 | May 11 2021

Task

The task is to predict the condition of a fetus based on the data received from a CTG machine.

DATA

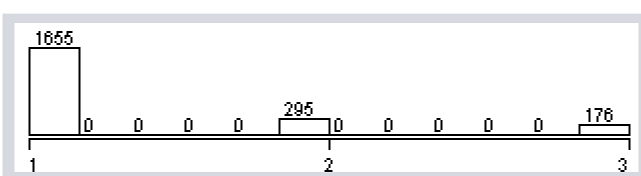
The data is downloaded from Kaggle. It has 21 features:

1. baseline_value
2. Accelerations
3. Fetal_movement
4. uterine_contractions
5. light_decelerations
6. severe_decelerations,
7. prolongued_decelerations,
8. abnormal_short_term_variability,
9. mean_value_of_short_term_variability
10. percentage_of_time_with_abnormal_long_term_variability,
11. mean_value_of_long_term_variability,
12. histogram_width,
13. histogram_min,
14. histogram_max,
15. histogram_number_of_peaks,
16. histogram_number_of_zeroes,
17. histogram_mode,
18. histogram_mean,
19. histogram_median,
20. histogram_variance,
21. histogram_tendency

Most of these features are continuous.

Each example is labelled as one of the three {1: Normal, 2: Suspect, 3: Pathological}

The class imbalance is high with a staggeringly high number of normal cases.



TEST TRAIN SPLIT

Since there is high class imbalance, the data is split into train and test data by keeping the proportion of classes same as in the given data.

METRIC FOR EVALUATION

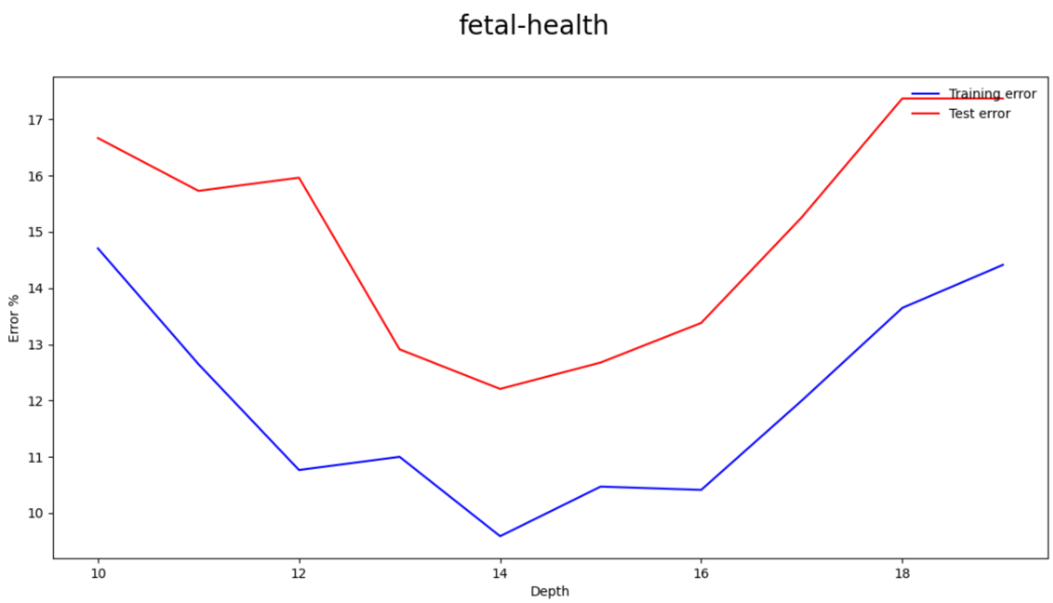
I have chosen to use the F1 score = 2(Precision)(Recall)/(Precision+Recall) as the evaluation metric for this multi class classification problem.

DECSISION TREES WITH BAGGING AND ADABOOST:

Preprocessing:

For decision trees, since attribute value pairs are needed, the first task is to discretize the features. I have implemented equal width binning with a bin size of 10.

To observe how decision trees are performing, the train and test error is observed for trees of varying depth.



It can be inferred that the most relevant information is learned from trees of depth around 14.

For depth 14, the F1 Score for the 3 classes is [0.92237443 0.592   0.74285714]

Bagging

Bagging is implemented on trees of depth 14 and 15 with bag size 10 and 20. There is noticeable improvement in the F1 score [0.92995529 0.61261261 0.85714286], especially in the classes 2 and 3.

Adaboost

Adaboost requires weak classifiers. So, decision tree with depth 10 is taken as base classifier. It perform poorly.

depth: 10

Base decision tree:

[0.90780142 0.56   0.46808511]

Boosting:

bag_size: 20 : [0.87450462 0.   0.   ]

bag_size: 30 : [0.87450462 0.   0.   ]

NEURAL NETWORKS

I implemented a neural network with 3 output nodes. Before passing it to the network, the data was split in the same way as above with 80% training data and 20% test data. The true labels are then one-hot encoded. The neural network has 3 layers with a sigmoid activation and mean square error loss function. The network was performing badly, which made me revise my loss function to a weighted mean square error, putting more weight to the 3rd class (pathological). However the classifier was entirely biased towards the 3rd class or towards the 1st class. It gave an F1-score of [0.87450462 0.   0.   ]. While it did give a good score once, most of the time it gave the above score. I think it means we need to find a better way to split and also compute the loss.

SVM

An SVM without kernel gives considerably good results. I implemented a 3-class SVM by running the SVM classifier 3 times for each class in a one-vs-all method. In test examples where there was more than one possible classification, I classify it as class 2, since it is the 'suspect' class, which suits best for non-confident classification. The F1 score for SV is [0.81583477 0.02272727 0.39344262] for lambda 0.1. For higher lambda values, class 3 is the only one classified correctly. The F1 score for the 2nd class is poor due to high number of false positives.

CONCLUSION:

It is observed that while decision trees of depth 14, Bagging and SVM gave good results, Neural nets and Adaboost did not. For neural nets, I haven't tried out other types of splits and loss functions. This is something I look forward to.

I expected Adaboost to give good results especially since the low number of class 3 examples shouldn't affect the performance as it penalizes incorrect classifications with greater weights.