Datathon Academic Report TM50

Author: (TM50.) **Hou Junning, Tian Sijia, Fang Mingzhi**

1. Data clean and process

The original dataset is provided in CTG.xls. We first import it as a pandas DataFrame and then export it as CSV for easier processing. We remove empty or invalid values and also create a scaled version for models sensitive to feature scale, such as logistic regression.

2. Data visualization and explore

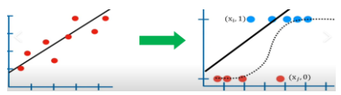
To simply approach the data set and see the relations of features first, we do some basic univariate and bi-variate visualization both on class data and numerical data. Include boxplot, histogram, violinplot, correlation heatmap.

3. Simple approach – Logistic Regression Model

After gaining preliminary insights into the relationships between features, we applied a simple logistic regression model:

$$z = w_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n = \mathbf{w}^T\mathbf{x} + b$$

$$p = \sigma(z) = \frac{1}{1+e^{-z}}$$



which adapts a linear model for classification tasks. Scaled data were used, as logistic regression is sensitive to feature magnitude. The trained model achieved an overall accuracy of 89.3%.

4. Complex and deeper approach ⭐

(1) Decision Tree

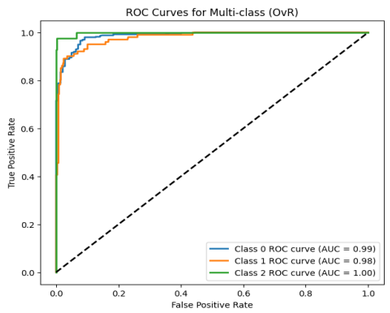
Considering the reason that Decision trees are highly interpretable, their decisions are transparent if–then threshold splits, which makes them easy to audit and explain, especially in domains like healthcare and finance. So we try to use it for predicting the NSP and what features have higher weight. We first set NSP as target and dropped it from the original column, then we fitted the data inside the decision tree model, and after running out the result(shown in picture), we can find that the decision tree has an accuracy of 0.89 and the top 3 features are Mean(0.25), MSTV(0.23) and ALTV(0.13).

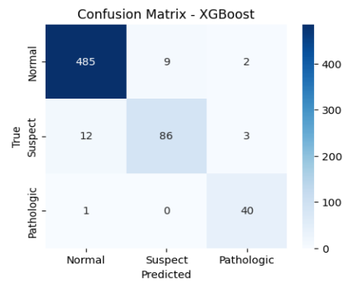
```
Accuracy: 0.8934169278996865
              precision    recall  f1-score   support

         1.0       0.98      0.88      0.93       496
         2.0       0.69      0.91      0.79       101
         3.0       0.67      1.00      0.80        41

    accuracy                           0.89       638
   macro avg       0.78      0.93      0.84       638
weighted avg       0.92      0.89      0.90       638

Confusion matrix:
 [[437  41  18]
 [  7  92   2]
 [  0   0  41]]
```

```
Top features:
 Mean        0.252500
MSTV         0.233082
ALTV         0.134926
ASTV         0.127828
AC           0.073964
Median       0.059191
DP           0.038090
Min          0.025120
Max          0.018288
Mode         0.013501
dtype: float64
```

(2) XGBoost

Xgboost was chosen as it performs well on structured tabular data like CTG features. It combines multiple decision trees to minimize prediction error through Boosting. For xgboost, each new tree focuses on correcting the errors made by previous trees by fitting the residuals. The final prediction is a weighted combination of all trees, regularized to prevent overfitting. We removed CLASS before training, since it was derived from the same medical annotations as the target variable NSP. After training and testing, the model achieved excellent performance on the cleaned CTG dataset, with an accuracy of above 95%. This results indicates that the xgboost model has a great advantage over traditional models.

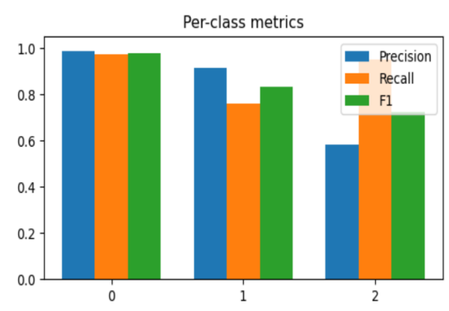




(3) Deep Learning (MLP)

Besides machine learning, we also tried deep learning to find out whether it has a better prediction on NSP. In real world, the signals (especially physiological signal) are often continuous instead of serrated steps(1), and small MLP has string ability learning smooth, non-linear decision boundaries that better match CTG features such as MSTV/ASTV/ALTV, so we choose the small MLP to predict the NSP. The process is straightforward: we first prepare x/y (cast to float32, relabel {1,2,3} to {0,1,2}), then compute inverse frequency class_weight, build a small MLP, then turn off 20% of the cells during training to prevent overfitting, let 64 cells to calculate and learn non-linear patterns, and then we split 20% of the validation as ‘tester’ to supervise the result, at last we use Earlystopping to stop learning if the validation stops improves for 10 rounds. After these operations, we can find that it has 0.93 of accuracy and performs well in the precision of 0 and 1(NSP 1 and 2).

```
Accuracy: 0.9373040752351097
              precision    recall  f1-score   support

           0       0.99      0.97      0.98       496
           1       0.92      0.76      0.83       101
           2       0.58      0.95      0.72        41

    accuracy                           0.94       638
   macro avg       0.83      0.90      0.85       638
weighted avg       0.95      0.94      0.94       638

Confusion matrix:
 [[482   6   8]
 [  4  77  20]
 [  1   1  39]]
```



[1]https://biomedical-engineering-online.biomedcentral.com/articles/10.1186/s12938-023-01075-1