A comparison of Decision Tree and k-Nearest Neighbours on Classification of Foetal Health

Sin Sze Li 210035720

12 December 2021

I. Brief Description and motivation of the problem

In this study, Decision Tree and k-Nearest Neighbours are used to solve a classification problem on foetal health. Using measurements from cardiotocograms, a foetus is classified as normal or abnormal. The performance of the two models are compared based on accuracy, recall, precision, F1 score, ROC curves, training and test time. Classifying foetal health by cardiotocograms measurements can help healthcare professionals identify irregular cases and take actions to prevent child and maternal mortality.

II. Initial Analysis of the Data Set including basic statistics

The dataset is 'Fetal Health Classification' available on Kaggle.

- The dataset contains measurements from cardiotocogram (CTG) results. There are 2126 samples.
- Each sample has 21 features. However, 10 of the columns are characteristics of histogram which do not bear any value or significance in the classification. Those columns are deleted so 11 features are used to train the two models, including foetal heart rate, foetal movements, uterine contractions and more.
- All the features are numerical.
- The original dataset has three target variables (classes): 'Normal', 'Suspect' and 'Pathological'. They are reclassified as 'Normal' and 'Abnormal' (which contains 'Suspect' and 'Pathological' cases), making this a binary classification problem.
- There is no missing value in the dataset.
- A class imbalance exists because there are 1655 normal samples and only 471 abnormal samples, as shown in Figure 1.
- The correlation heatmap (Figure 2) indicates the correlation between each feature and class. There is a strong negative correlation between accelerations and foetal health, meaning acceleration of the foetus indicates healthy foetus.
- The strip plots in Figure 3 show the distribution of each feature for both classes, which can also help analyse the correction heatmap. For example, in the 'accelerations' strip plot, normal foetuses do have larger accelerations than abnormal foetuses, which corresponds to the finding in the previous point.

	mean	std	min	50%	max
baseline_value	133.303857	9.840844	106.0	133.000	160.000
accelerations	0.003178	0.003866	0.0	0.002	0.019
fetal_movement	0.009481	0.046666	0.0	0.000	0.481
uterine_contractions	0.004366	0.002946	0.0	0.004	0.015
light_decelerations	0.001889	0.002960	0.0	0.000	0.015
severe_decelerations	0.000003	0.000057	0.0	0.000	0.001
prolongued_decelerations	0.000159	0.000590	0.0	0.000	0.005
abnormal_short_term_var	46.990122	17.192814	12.0	49.000	87.000
abnormal_short_term_var_mean	1.332785	0.883241	0.2	1.200	7.000
abnormal_long_term_var	9.846660	18.396880	0.0	0.000	91.000
abnormal_long_term_var_mean	8.187629	5.628247	0.0	7.400	50.700

Table 1. Statistical summary of features

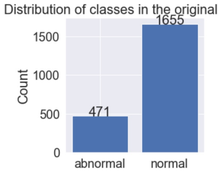

Figure 1. Bar plot of classes

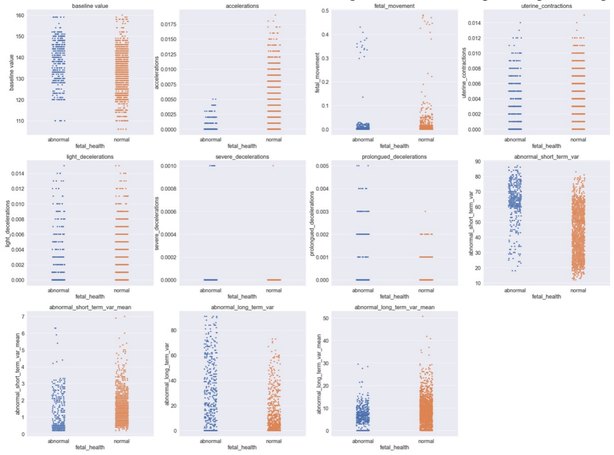

Figure 3. Strip plots of each feature

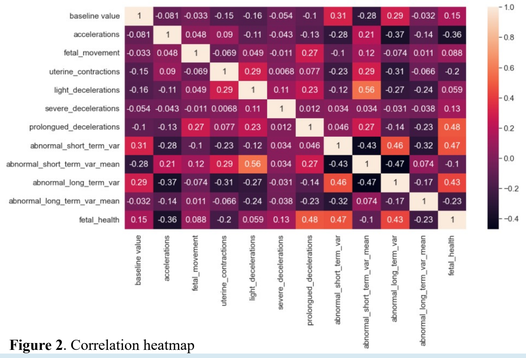

Figure 2. Correlation heatmap

III. Brief summary of the two ML models with their pros and cons

1. Decision Tree (DT)

How it works

- DT is a supervised learning method to predict the target variable based on decision rules inferred from the features.
- A tree consists of nodes, branches and leaves. The topmost root node tests on a feature, which then leads downwards to internal nodes via branches based on different possible feature values against a threshold. This process is repeated until a terminal node, a leaf, is reached. Each leaf assigns a class.

Pros

- Easy to visualise
- Simple to understand and interpret
- Normalization and scaling of data is not required
- Both continuous and categorical variables can be used
- Missing values are automatically handled
- Requires little data preparation
- Robust to outliers
- Can be used for both classification and regression problems

Cons

- Overfitting occurs easily
- Adding new datapoints leads to retraining the model
- Easily affect by noise
- Reduced performance when the training set is small [1]
- Not suitable for large datasets because the tree could require exponentially many nodes

2. k-Nearest Neighbour (kNN)

How it works

- kNN is a supervised learning method which assumes similar data points exist in close proximity.
- k is the number of nearest neighbours. When deciding the class of a sample, classes of the k closest points to the sample are found. Each neighbour votes for their class, and the most votes is taken as the prediction.

Pros

- Easy to implement because there is only two hyperparameters, value of k and the distance function
- No training step is needed – kNN does not explicitly build a model, because it learns from the existing data only when making the prediction
- Since no training step is needed, new data can be added to kNN seamlessly without retraining the model
- Nonparametric, so no assumption is made about the data
- Can be used for both classification and regression problems

Cons

- Computation cost is higher than DT for large dataset because the distance between the sample and each existing points are calculated
- Sensitive to outliers and data with noise
- Curse of dimensionality – kNN performs better with a lower number of feature than a large number of features
- Easily affected by class imbalance. Since the prediction of a sample relies on the number of vote of each class by the neighbours, the majority of the classes would likely be the final prediction

IV. Hypothesis statement

Literature review[3] [5] shows that DT slightly outperformed kNN on the foetal health dataset. It is therefore expected that DT will perform better than kNN in terms of accuracy and F1 score. Moreover, as kNN has a higher computational cost than DT for large dataset, the training and testing time for kNN might be longer than DT.

V. Description of the choice of training and evaluation methodology

- The dataset is split into 70% training set and 30% test set with stratification. The test set is kept unseen until the final comparison of the two models.
- Because of the class imbalance, Synthetic Minority Oversampling Technique (SMOTE) is applied to the training set. As a result, there are equal number of 'normal' and 'abnormal' classes in the training set, as shown in Figure 4.
- After the splitting and applying SMOTE, there are 2316 training data and 638 test data.
- For each of the model, Bayesian optimization with 10-fold cross validation is applied on the training data to find the best hyperparameters. The optimization process is based on the F1 score of the models.
- Run the models on the test set. To compare the performance of the models, confusion matrix, accuracy, recall, precision, F1 score, ROC and AUC are computed.

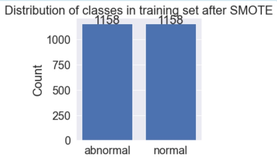

Figure 4. Bar plot of classes after SMOTE

VI. Choice of parameters and experimental results

1. Decision Tree

- Choose split features using standard CART algorithm with Gini's Diversity Index as split criterion.
- Control the depth of the tree by optimizing the maximal number of decision splits ('MaxNumSplits') and minimum number of leaf node observations ('MinLeafSize').

From the result of Bayesian optimization:

- MaxNumSplits = 67
- MinLeafSize = 20

2. k-Nearest Neighbour

- Use the exhaustive search algorithm to find the nearest neighbours by calculating the distance values from all existing point to each point to be classified.
- Optimize the model by running Bayesian optimization on Number of Nearest Neighbours to find ('NumNeighbors') and Distance metric ('Distance').

From the result of Bayesian optimization:

- NumNeighbors = 15
- Distance = 'cityblock'

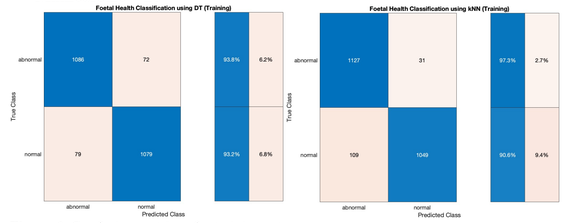

Figure 6. Confusion matrices for training set

DT	Evaluation metric	kNN
0.908	Accuracy	0.903
0.879	Recall	**0.965**
0.747	Precision	0.704
0.808	F1 score	**0.814**
0.951	ROC	**0.981**

Table 2. Performance on test set

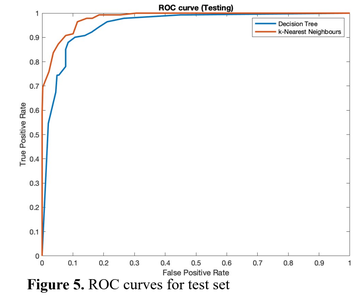

Figure 5. ROC curves for test set

Note that 'abnormal' is considered as the positive class, while 'normal' is considered as the negative class.

VII. Analysis and critical evaluation of results

- In clinical practice, once abnormality is identified, follow-up action must be taken to ensure the safety of the foetus and the mother. Therefore, the risk of having false negative cases is high. However, if a normal foetus is wrongly classified as abnormal, the mother would be worried about the baby and her mental health would be greatly affected. Therefore, F1 score is chosen to be the main evaluation metric in this study.
- In terms of accuracy, recall, precision, F1 score and ROC, DT and kNN have similar performance. Accuracy is almost the same (0.908 and 0.903). They also have very similar recall (0.879 for DT and 0.965 for kNN), F1 score (0.808 for DT and 0.814 for kNN) and AUC (0.951 for DT and 0.981 for kNN).
- SMOTE generates new datapoints nearby for the minority class regardless of the class of neighbouring examples. As a result, the new synthetic datapoints may overlap existing datapoints from other classes and introduce noise to the dataset, which in turn affects the performance of both models.
- The slightly better accuracy for DT may be because of its working principle. Both models are trained by noise-introduced training set. DT is trained based on certain thresholds of the features, while kNN is based on the the distances between datapoints. Although the noises in synthetic data may affect the chosen threshold in DT, once the thresholds are determined, the noises are 'absorbed' into either side of the threshold. In kNN, when the distances are calculated, those noises are also captured to determine the final class of the sample. As a result, the effect of noises in kNN is more significant than in DT.
- With the test set, both models have higher recall than precision. However, recall and precision are equally high for both models with training set. This indicates that both models are overfitted, and they have high variance and low bias.
- The problem of higher recall than precision is also due to the class imbalance in the test set. Compare Figure 6 and Figure 7, in training set there is no class balance so the number of false positive and false negative are similar. In test set, since negative class is the majority, the number of false positive is much higher than false negative. As recall puts false negative in the denominator and precision puts false positive in the denominator, precision will be lower than recall when the dataset is skewed towards the negative class.
- In kNN, the features must be normalised before training and testing. In this study, they are rescaled to the range 0 to 1. This is because kNN relies on the distance between datapoints so all features should be on the same scale. This is not required for DT.
- Training time (Bayesian optimization) was 4.36 s for DT and 3.76 s for kNN, which is contradictory to the hypothesis, although a difference of less than 1 s is insignificant in this case. Both of them have testing time of within 0.2 s.

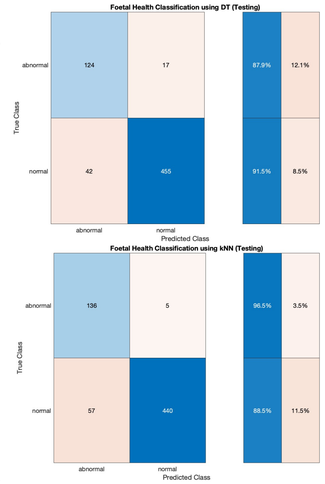

Figure 7. Confusion matrices for test set

VIII. Lesson learned and future work

- Data pre-processing is important. Besides missing values, the problems of outliers and noises should also be addressed properly.
- Class imbalance plays a crucial role in model performance. Combining SMOTE with other under-sampling techniques such as Tomek Links and Edited Nearest Neighbours (ENN) may improve model performance.
- ROC may be less informative for model comparison with imbalance datasets. The false positive rate tends to be small due to the large number of negative class. Using Precision-Recall (PR) curve to reflect model performance may be more suitable.
- Feature selection can be applied to the dataset to reduce the number of inputs that are irrelevant to the model. Feature selection method includes ANOVA and Kendall's.
- Other optimization algorithm such as grid search or random search can be tried to further optimize the models. Grid search takes a longer time than Bayesian optimization because it exhaustively searches for every combination of hyperparameters but it may result in better optimization.

IX. References

[1] Katz, G., Shabtai, A., Rokach, L. and Ofek, N. (2014). ConfDTree: A Statistical Method for Improving Decision Trees. *Journal of Computer Science and Technology*, 29(3), pp.392–407.

[2] Kubat, M. (2018). *Introduction To Machine Learning.* S.L.: Springer International Pu.

[3] Li, J. and Liu, X. (2021). Fetal health Classification Based on Machine Learning. *2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering.*

[4] Mehbodniya, A., Lazar, A.J.P., Webber, J., Sharma, D.K., Jayagopalan, S., K, K., Singh, P., Rajan, R., Pandya, S. and Sengan, S. (2021). Fetal health classification from cardiotocographic data using machine learning. *Expert Systems*.

[5] Noor, N., Ahmad. N. and Noor, N. (2021). Fetal Health Classification Using Supervised Learning Approach. *2021 IEEE National Biomedical Engineering Conference.*

[6] Piri, J. and Mohapatra, P. (2019). Exploring Fetal Health Status Using An Association Based Classification Approach. *2019 International Conference on Information Technology.*