ASEAN Engineering Journal

Full Paper

AN EMPIRICAL ANALYSIS OF FEATURE ENGINEERING TECHNIQUES TO REDUCE DIMENSIONALITY FOR NON-BINARY CLASSIFICATION PROBLEMS - A CASE STUDY WITH FOETAL HEALTH DATASET

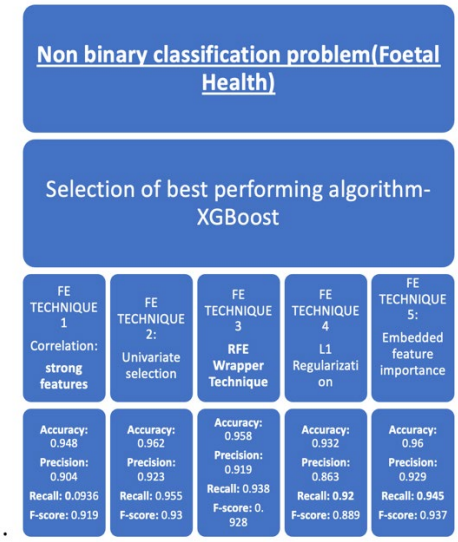
Sandhya Soman[a*], Adeitia Kalyann Boniface[b*], Agnes Lydia[c]

[a]GITAM(Deemed-to-be) University, School of Science, Department of Computer Science, Bengaluru, India
[b]Indian Institute of Management, Vishakhapatnam, Andhra Pradesh, India.
[c]AI/ML Associate Consultant, Sustainable Living Lab, Chennai, India

Article history
Received
19 June 2024
Received in revised form
03 October 2024
Accepted
20 November 2024
Published online
31 August 2025

*Corresponding author
ssoman@gitam.edu

Graphical abstract



Abstract

In today's world, when AI and ML are deeply involved in our day-to-day lives, merely designing a machine learning model is insufficient. The complexity involved in training the model is vital in determining whether such systems would be deployed in real environments. Today, ML engineers strive to accomplish this, as the models that work well in academic research fail to work well in production. The ML code is a small segment of the ML infrastructure. While in Academia, the focus is on code and hyperparameters, the Industrial Product Team's focus is on data. Data engineering and feature engineering, often ignored during model creation and deployment, are two techniques to bridge this gap. To emphasize its importance, we have considered a non-binary classification problem - the Foetal Health Classification problem. We have applied different feature engineering techniques to reduce the number of significant features required for Model Training and have determined the best possible FE technique. From a set of 21 independent features, we could lower the feature count to nine and retain the accuracy score compared to training using the complete feature set. This paper showcases the performance of different prediction models on the dataset, selecting the best prediction model and applying feature engineering techniques for dimensionality reduction. Keeping the threshold at 0.025, we could achieve 96% accuracy, 92.9% precision score, 94.5% recall value, 93.7% F-score, and a dimensionality reduction of 29%. Maintaining a threshold of 0.013, a 95.1% accuracy, 91.3% precision value, 94.5% recall, and 92.8% F-score, and a dimensionality reduction of 57% could be achieved. The above indicates that equivalent results can be achieved with a subset of the Feature set, which can be further instrumental in reducing the model training and convergence time.

Keywords: Dimensionality Reduction, Feature Subspace, Feature Reduction, Feature Engineering, Convergence Time

1.0 INTRODUCTION

In today's world, there is no dearth of data. Although the volume is enormous, how much we can make sense of it and use it to solve a business problem is constantly debated. A machine learning algorithm's performance depends on the input data it consumes. Irrespective of whether the data is labelled, the quality of the input determines the quality of the model's output. The attributes of the data are called features [1]. It has been proven in the literature that the model performance is not dependent on all the features but is based on a few relevant ones. Feature engineering is a step that is used to uncover those relevant features. However, it is often ignored in most ML projects due to the additional time requirements for this step [2].

As the interest grew in applying ML-based solutions for real-world problems, it was observed that while the solutions performed well in academic research, several issues were faced when deployed on a larger scale. In this regard, feature

engineering techniques' efficient and effective usage can be significant. Some relevant and related studies that have used ML techniques and applied feature selection are discussed in the following paragraphs.

One of the exciting works in this area is done by Tim et al. [5]. They stated that to improve any ML model's performance, the focus should be on data rather than creating new algorithms. The author phrases it as "clever engineering", which is particularly beneficial for predictive models. Uddin et al. support finding an "optimal dataset" for optimisation in ML. They have proposed a technique for adding and removing features in parallel during training [4]. On the contrary, Jundong et al. [6] advocate that feature selection makes an ML model more comprehensible. They have presented a comprehensive list of open-source feature selection algorithms. A straightforward line of approach has been proposed by Fernandez et al. [7]. Their work addresses the determination of which feature selection techniques suit a given problem. In their work, they have strongly advocated using correlation-based feature selection as one of the most potent tools in feature engineering for dimensionality reduction. A study by Jebadurai et al. [21] used filtering-based feature selection approaches to increase classification accuracy by 4%. Regmi & Shah[22] achieved the maximum accuracy of 94.36 by using TabNet. However, they have used only three algorithms: Random Forest and TabNet.

A key observation, which was also the primary motivation for this study, is that in academic research, we always focus on tuning the model's hyperparameters to get better evaluation metrics with less emphasis on the data [3]. As a result, in a real production environment, such ML projects do not perform well or do not perform as expected. This is a serious problem in the current scenario, considering the large amount of money and resources invested in academic research [4].

Considering the above-mentioned gap, this study aims to understand the impact of feature engineering techniques on the overall performance of various classification algorithms and whether it is worth investing time and effort into. To exemplify the significance of Feature Engineering and its potential benefits, we have considered a non-binary classification problem - Foetal Health Classification, where the records are classified as either healthy, suspicious, or diseased. We applied several classifiers and determined the number of features required to get the best evaluation metrics for them. We then selected the best classifier, i.e., XGBoost, for the current dataset. We then tried to reduce the dimensionality without impacting the evaluation metrics. We could lower the feature count from 21 to 9 and achieve a comparable evaluation metric.

The following sections will discuss our methodology and findings in detail. The rest of the paper is divided as follows: Section 2 discusses the dataset and methodology used. Sections 3 discuss the significant results. The conclusion and findings are listed in Section 4.

2.0 METHODOLOGY

In this paper, we have approached feature selection from a data perspective. The methodology adopted is displayed in Figure 1. Four main steps are involved: data cleaning and transformation, data distribution, model prediction and cross-validation, and model selection.

2.1 Data Cleaning and Transformation

The dataset was scrutinised for null/missing values; however, all features were retained as there were no null values. Also, there were no categorical values, so no encoding was performed on the dataset. The records were standardised using Standard Scaling.

2.2 Data Distribution

The records of the dataset belong to three classes: Normal (1), Suspect (2), and Pathological (3). The given problem is a multi-class classification problem. The dataset is imbalanced with 1655 records belonging to Class 1, 295 to Class 2, and 176 to Class 3. To balance the dataset, we used SMOTE to generate synthetic samples for minority classes.

2.3 Model Prediction, Cross-validation and Model Selection

For the given classification problem, we created a pipeline from six classifiers: Logistic Regression, Decision Tree, SVC, Random Forest, KNN, and XGBoost. Also, to increase the assurance of the model's performance, we performed stratified k-fold, k-fold, and cross-validation on each model and selected the best representative.

The best algorithmic model determined from the previous step was then subjected to five different feature engineering techniques - Filter methods (strong features correlation, univariate feature selection), one Wrapper method (recursive feature elimination), and two Embedded methods (L1 regularisation and feature importance methods, to see if there was an improvement in the metrics or no difference at all [13]. We observed that we could retain accuracy and get better precision, recall, and F1 scores by reducing the feature variables. We could attain a reduction of up to 9 features, which will be discussed in detail in the following sections.

The following subsection discusses the Dataset Used in detail.

2.3.1 Datasets Used

For this paper, we have utilised the Foetal Health Classification dataset from the UCI ML repository [8][9]. This dataset contains the CTG readings of the foetus [10]. Cardiotocography recordings are used to monitor the heart rate of the foetus [1][11] and to detect the possibility of complications in pregnancy. The standard heart rate is between the range of 110 to 160 bpm.

The dataset captures the following [12]:

- PM1: the base value for the number of heartbeats per minute.
- PM2: the increase in the foetal heart rate.
- PM3: the number of foetal movements in a second.
- PM4: the number of uterine contractions in a second.
- PM5: the decrease in the foetal heart rate (mild).
- PM6: the reduction in the foetal heart rate (severe).
- PM7: the decrease in the foetal heart rate (prolonged).
- PM8: the reduction in the foetal heart rate (mild).
- PM9: the % of the time with abnormal variations (short term) in the foetal heart rate.
- PM10: the mean value of PM9.

- PM11: the % of the time with abnormal variations (long term) in the foetal heart rate
- PM12: the mean value of PM11.
- PM13: the heart rate's histogram's width.
- PM14: the min value of the histogram.
- PM15: the max value of the histogram.
- PM16: the total number of peaks in the histogram.
- PM17: the total number of zeros in the histogram.
- PM18: histogram's mode value.
- PM19: histogram's median value.
- PM20: histogram's variance.
- PM21: histogram's tendency for data distribution.
- PM22: foetal health.

PM22 is the target variable, and PM1-PM22 are treated as independent variables. The status of the foetus's health is specified using three classes- class 1, class 2, and class 3. Class 1 represents the 'normal case'/healthy. Class 2 designates a 'suspect case', i.e., some abnormality or disease is possible. Class 3 represents a 'pathological case' or diseased condition. The dataset has an imbalance and contains 1655 records for class 1, 295 for class 2, and 176 for class 3. We have used SMOTE to generate synthetic records.

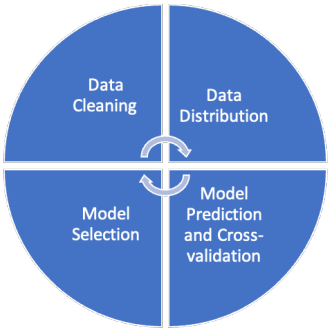


Figure 1 Research Methodology

3.0 RESULTS AND DISCUSSION

The Foetal Health dataset considered in this paper is a classic example of a non-binary classification. Twenty-two features are present in the dataset. All the features were explained in Section 3. We have 21 independent features (PM1 - PM21) and one dependent variable, PM22 (FOETAL HEALTH).

The following section discusses the model performance obtained using all 21 features, to which feature engineering was later applied to get the same/enhanced performance metrics with fewer features. In the first phase, the metrics were obtained for each model on the original, scaled, and augmented datasets (SMOTE) and scaled and augmented datasets (SMOTE and scaled) [14].

3.1 Logistic Regression Model

The cross-validation accuracy scores obtained for the logistic regression model are given in Table 1.

Table 1 Accuracy Scores - Logistic Regression Model

Techniques	LR	LR-Scaled	LR-Smote	LR Smote + Scaled
K-Fold	0.8800	0.8953	0.8276	0.8938
STK-Fold	0.8776	0.8947	0.8268	0.8959
CV=10	0.8824	0.8959	0.8255	0.8941

The model gave the best results for Stratified K-fold with an accuracy of 89.6%. The model performs best with scaling compared to base LR and hives a better performance of 0.8959 using Stratified k-fold. This could be because scaling allows better model calibration and reduces the influence of feature magnitudes, resulting in better predictions. The test accuracy score is given in Table 2. The SMOTE+Scaled model with STK-fold gives a satisfactory accuracy of 0.84 and precision notably higher than the recall value, suggesting that the model can handle most positive cases. The F-score of 0.854 indicates a well-calibrated model.

Table 2 Scores - Logistic Regression

Metrics	Stratified K-fold
Accuracy	0.840
Precision	0.883
Recall	0.840
F-score	0.854
Validation	STK-Fold
Data	Smote+Scaled
No. of features	21

3.2 Decision Tree Model

The cross-validation accuracy scores obtained for the decision tree model are given in Table 3.

Table 3 Accuracy Scores - Decision Tree

Techniques	DT	DT-Scaled	DT-Smote	DT Smote + Scaled
K-Fold	0.9112	0.9112	0.9581	0.9579
STK-Fold	0.9118	0.9135	0.9592	0.9592
CV=10	0.9118	0.9135	0.9597	0.9597

The model gave the best results for 10-fold cross-validation. The test accuracy score is shown in Table 4.

Table 4. Scores - Decision Tree

Metrics	10-fold CV
Accuracy	0.925
Precision	0.93
Recall	0.925
F-score	0.927
Validation	CV = 10
Data	Smote+Scaled
No. of features	21

The model performs well with the given dataset. The highest accuracy is achieved in the Smote+Scaled model, which allows it to generalise across minority classes. The test accuracy indicates that the model works well in identifying positive cases and is a robust predictor for the given dataset.

3.3 Random Forest Classifier

The cross-validation accuracy scores obtained for the random forest model are given in Table 5. The highest accuracy is yielded in the combination of Smote + Scaled, with almost similar results of 0.9798 and 0.9791 for k-fold and stk-fold, with STK-fold marginally outperforming k-fold

Table 5 Accuracy Scores - Random Forest Model

Techniques	RF	RF-Scaled	RF-Smote	RF Smote + Scaled
K-Fold	0.9400	0.9376	0.9796	0.9798
STK-Fold	0.9435	0.9441	0.9786	0.9791
CV=10	0.9347	0.9347	0.9793	0.9793

The model gave the best results for Stratified K-fold. The model test accuracy score is shown in Table 6. Test results indicate a well-balanced model with precision and recall and also offer a robust generalisation with an f score of 0.944

Table 6 Scores – Random Forest

Metrics	STK-Fold
Accuracy	0.944
Precision	0.946
Recall	0.944
F-score	0.944
Validation	K-Fold
Data	Smote+Scaled
No. of features	21

3.4 K- Nearest Neighbor

The following cross-validation accuracy scores were obtained for the KNN with 3,5,6,9,11,13, and 15 neighbours. The best scores for the Stratified K-fold were obtained, as shown in Table 7. The results indicate variation with the number of neighbours, with the highest accuracy with k =3. The performance declines with increasing neighbours, showing that smaller k values best suit the foetal dataset.

Table 7 Accuracy - Comparison between the Models for Stratified K-fold

Neighbors	KNN	KNN Scaled	KNN Smote	KNN Smote+ Scaled
3	0.9041	0.9006	0.9483	0.9624
5	0.8965	0.8924	0.9425	0.9513
7	0.8918	0.8959	0.9385	0.9433
9	0.8876	0.8929	0.9276	0.9392
11	0.8812	0.8941	0.9211	0.9339
13	0.8741	0.8906	0.9176	0.9276
15	0.8735	0.8847	0.9118	0.9254

The test accuracy score for KNN is given in Table 8. The results indicate a strong balance between precision and recall, achieving a satisfactory accuracy of 0.904.

Table 8 Scores - K- Nearest Neighbor

Metrics	STK-Fold
Accuracy	0.904
Precision	0.918
Recall	0.904
F-score	0.908
Validation	STK-Fold
Data	Smote+Scaled
No. of features	21

3.5 Support Vector Classifier

The following cross-validation accuracy scores obtained for the Support Vector Classifier model are given in Table 9. Like previous algorithms, SVM also performs best in SMOTE+Scaled form when addressing the model's ability to handle class imbalance. It shows the best result of 0.9433 with 10-fold cross-validation.

Table 9 Accuracy Scores - Support Vector Classifier Model

Techniques	DT	DT-Scaled	DT-Smote	DT Smote + Scaled
K-Fold	0.8541	0.9106	0.8477	0.9418
STK-Fold	0.8500	0.9071	0.8447	0.9415
CV=10	0.8571	0.9129	0.8470	0.9433

The evaluation metric for the test dataset for SVC is given in Table 10. SVM exhibits good metrics in the test dataset with appropriate preprocessing and optimisation; however, compared to other models, the results are not as good as those of different classifiers.

Table 10 Scores - Support Vector Classifier.

Metrics	STK-Fold
Accuracy	0.880
Precision	0.915
Recall	0.880
F-score	0.890
Validation	K-Fold
Data	Smote+Scaled
No. of features	21

3.6 XGBoost Classifier

The cross-validation accuracy scores obtained for the XGBoost Classifier model are given in Table 11. The classifier provides the best performance among the rest with the given dataset. For XGBoost also, feature scaling and SMOTE give the best accuracies, providing similar scores for k-fold(0.9829) and stk-fold (0.9808)

Table 11 Accuracy Scores - XGBoost Classifier Model

Techniques	XGB	XGB-Scaled	XGB-Smote	XGB Smote+ Scaled
K-Fold	0.9476	0.9476	0.9829	0.9829
STK-Fold	0.9512	0.9512	0.9808	0.9808
CV=10	0.9412	0.9412	0.9839	0.9839

The evaluation metric for the test dataset for the XGBoost Classifier obtained is given in Table 12. The test dataset scores also indicate a well-balanced model, excelling in precision and recall with an accuracy of 0.96. This performance is due to the advanced boosting algorithm, which works towards the iterative reduction of errors from previous steps.

Table 12 Scores - XGBoost

Metrics	STK-Fold
Accuracy	0.96
Precision	0.933
Recall	0.939
F-score	0.936
Validation	K-Fold
Data	Smote+Scaled
No. of features	21

3.7 Comparing Classifier Results

The following table (Table 13) summarises the results obtained from various classifiers. XGBoost has outperformed the others with an accuracy of ~96% on the test dataset. We have also considered the RECALL score as the next significant metric since the current classification problem belongs to the medical domain, and we aimed to prevent the number of false negatives, i.e., to reduce the number of misclassifications from CLASS 3 (DISEASED) to CLASS 2 (MAY BE DISEASED) or CLASS 1 (HEALTHY). XGBoost gives a comparable score of ~96%, the best score among the given classifiers, as in Table 13. The ability to prevent overfitting and to leverage gradient boost makes it the best-performing model for the foetal dataset considered for this study.

Table 13 Comparison of Different Classifiers – For 21 Features

Index	Accuracy	Precision	Recall	F-score	Validation
LR	0.840	0.883	0.840	0.854	STK-Fold
DT	0.925	0.930	0.925	0.927	CV=10
RF	0.944	0.946	0.944	0.944	K-Fold
KNN	0.904	0.918	0.904	0.908	STK-Fold
XGB	0.960	0.933	0.939	0.936	K-Fold
SVC	0.880	0.915	0.880	0.890	K-Fold

3.8 Feature Engineering

In the first phase of our research, we tried to find the best classification algorithm for foetal health detection; we received results that resonated with many of the earlier works published in this area, namely [1], [10], and [2].

In the next phase of this work, we tried applying five feature engineering techniques—two filter methods (strong features correlation, univariate feature selection), one Wrapper method (recursive feature elimination), and two Embedded methods (L1 regularisation and feature importance methods) [15][16] —to determine if we could get the same evaluation metric or better measures with fewer features.

The results obtained for each FE technique [17] [18] are as follows:

i. *Strong Feature Correlation*: In this section, we have used a supervised feature selection technique to extract features with a high correlation with the target variable. We have used Pearson Correlation, keeping a threshold greater than 0.2. This method performs the ranking operation irrespective of the model used for training. The following features were not considered-fetal_movement, light_decelerations, histogram_width, histogram_min, histogram_max, histogram_number_of_peaks, histogram_number_of_zeroes, as they had low correlation with the target variable "fetal health". We then trained the model with those features, and the values in Table 14 are the resultant evaluation metric.

Table 14 Scores - XGBoost-strong features

Metrics	XGBoost-strong features
Accuracy	0.948
Precision	0.9040
Recall	0.0936
F-score	0.919
Validation	K-Fold
No. of features	14

The elimination of 7 features has resulted in a dip of 1.25%, 3.11%, 0.32%, and 1.81% in accuracy, precision, recall, and F1 score, respectively.

ii. *Univariate Feature Selection*: In this technique, using the ANOVA f-test, we selected the top 17 features (k=17) after normalisation and evaluated the model based on these metrics [19][20]. The following features were not considered under univariate feature selection-histogram_number_of_peaks, severe_decelerations, histogram_width, and histogram_max. The main reason for exclusion could be that they may not be adding to the additional predictive power. For example, Histogram_number_of_peaks is a redundant feature that could have already been captured in mean or variance. Similarly, severe_decelerations would be redundant in the presence of prolonged_decelerations. Weak correlation with the target variable resulted in eliminating features like histogram_width and histogram_max, compared to other histogram features.

The scores below, as in Table 15, are representative when k=17. A further reduction in the k values resulted in a dip in accuracy and precision.

Table 15 Scores - Univariate Feature Selection

Metrics	XGBoost-univariate
Accuracy	0.962
Precision	0.923
Recall	0.955
F-score	0.938
Validation	K-Fold
No. of features	17

Applying the F-test resulted in an increase of 0.21% in Accuracy, a decrease of 1.1% in Precision value, and 1.7% in the Recall score, and a 0.21% increase in F-score with a feature count of 17 in a dataset of 2126 records.

iii. *Feature Subset using RFE Wrapper Technique*: RFE is a technique that begins with all features and recursively eliminates the features that do not improve the results. RFE can be used as a Wrapper technique around the model of our choice. We wrapped it around the XGB classifier, keeping the limit for feature count as 17. The eliminated features include light_decelerations, severe_decelerations, mean_value_of_long_term_variability, histogram_width. The feature light_decelerations has a weak association with adverse fetal outcomes. Severe_decelerations, from a model's perspective, might represent a redundant

feature when prolonged_ decelerations are already considered. There could be two reasons for the non-consideration of mean_value_of_long_term_variability. First, it is a redundant attribute in the presence of a percentage of time with abnormal long-term variability, and it is not sensitive to immediate changes happening to foetal health conditions. Lastly, the range of heart rate values captured by histogram_width might not be as informative as features like variance, which could be the reason for its non-inclusion in the model's final list.

The evaluation metric obtained is given in Table 16. There was a 0.2%, 1.5%,0.11%, and 0.8% dip in the accuracy, Precision, Recall, and F1 scores, respectively.

Table 16 RFE Wrapper Technique

Metrics	XGBoost-RFE
Accuracy	0.958
Precision	0.919
Recall	0.938
F-score	0.928
Validation	K-Fold
No. of features	17

iv. *L1–Regularization*: L1 Regularization eliminates the most minor significant(least) features. This works by attaching a penalty to the loss function. The seven features considered include: 'accelerations', 'prolongued_decelerations', 'abnormal_short_term_variability', 'percentage_of_time_with_abnormal_long_term_variability','histogram_mode', 'histogram_mean', 'histogram_median'. The possible reasons for inclusion are explained in the following few statements. The feature 'accelerations' is a critical predictor and indicator for feature health. The presence of 'prolongued_decelerations' and 'abnormal_short_term_variability' are clinical markers for medical intervention and can indicate fetal distress. The feature 'percentage_of_time_with_abnormal_long_term_variability' complements the feature 'short_term_variability' and could be used by the model for long-term fetal health prediction. The summary statistic measures-'histogram_mode', 'histogram_mean', and 'histogram_median' provide a comprehensive view of heart rate distribution, making them critical indicators. Features that were not selected provided minimal new information and, hence, had lesser predictive powers, such as baseline values, fetal movement, uterine contractions, and finer histogram details.

L1 Regularization results in Table 17 are obtained with the XGB Classifier, which applies an L1 penalty and a count of 100 estimators. There was a 2.9%, 7.5%,2.02%, and 5.02% dip in the accuracy, Precision, Recall, and F1 scores, respectively. This cannot be a preferred FE method for the given dataset.

Table 17 L1 Regularization

Metrics	XGBoost-L1
Accuracy	0.932
Precision	0.863
Recall	0.920
F-score	0.889
Validation	K-Fold
No. of features	7

Embedded Feature Importance Method: In this method, we used the model's intrinsic ability to select and use the most important features while it was being created. The model chooses the features that have a more significant impact on the dependent variable. All tree-based models, like Random Forest XGB Classifier, rank features based on that and work on the most important features. When using feature importance, considering the feature importance graph, we considered two threshold values- 0.025 and 0.013, based on the information presented in Figure 2. The selected features combine the rate variability measures, central tendency and extreme values, which provide a 360-degree view of fetal health, making them essential predictors for fetal health outcomes.

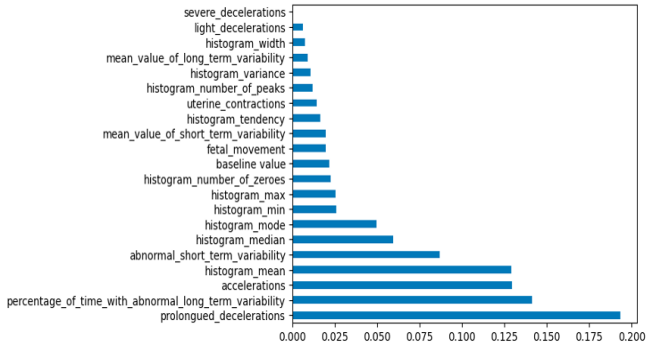


Figure 2 Feature Importance Graph

Table 18 lists the features shortlisted by XGBoost, and Table 19 shows those selected by the embedded method at a threshold of 0.025

Table 18 XGBoost- Shortlisted Features

Sl. No	Features
1	'accelerations'
2	'prolongued_decelerations'
3	'abnormal_short_term_variability'
4	'percentage_of_time_with_abnormal_long_term_variability'
5	'histogram_min'
6	'histogram_max'
7	'histogram_mode'
8	'histogram_mean'
9	'histogram_median'

Table 19 XGBoost - Embedded Methods(th=0.025)

Metrics	XGBoost-Embedded Methods
Accuracy	0.951
Precision	0.913
Recall	0.945
F-score	0.928
Validation	K-Fold
No. of features	9

As is evident in the results, we could reduce the number to 9 (~57% decrease). The elimination of 12 features has resulted in almost the same accuracy, 95.1%, compared to all 21 features (96%). We could achieve a higher recall (0.6%). This is a significant result in the given classification problem, as we could achieve the same accuracy with fewer features and greater recall value, which is particularly helpful with false negatives. The results would be of substantial use in large datasets.

The results in Table 20 were obtained when the threshold was kept at 0.013. The features considered at this threshold includes 'baseline value', 'accelerations', 'fetal_movement', 'uterine_contractions', 'prolongued_decelerations', 'abnormal_short_term_variability', 'mean_value_of_short_term_variability', 'percentage_of_time_with_abnormal_long_term_variability','histogram_min','histogram_max','histogram_number_of_zeroes', 'histogram_mode', 'histogram_mean', 'histogram_median', 'histogram_tendency'

Table 20 Embedded Methods(th=0.013)

Metrics	XGBoost
Accuracy	0.960
Precision	0.929
Recall	0.945
F-score	0.937
Validation	K-Fold
No. of features	15

We could bring down a 29% decrease in the feature count with 15 features for this value. Even with the elimination of 6 features, we could increase accuracy. We could reach a score of 96.2%, a slightly higher F1 score (93.6%), and a higher recall value (0.6%). We could achieve the same accuracy and higher recall value with fewer features. The results would have a higher impact on large datasets.

4.0 CONCLUSION

Bringing down the number of features used for model training can significantly impact the training time and, thus, the model performance, especially when dealing with terabytes of data. Through this paper, we have applied five popular techniques and evaluated their result on the Foetal Health Classification dataset. We could observe that we could retain accuracy and get better precision, recall, and F1 scores by reducing the feature variables. We could attain a reduction of up to 9 features. This can be instrumental to a decrease in training time, especially with massive datasets.

Table 21 summarises the accuracy (AC), precision (PR), recall (RC), F- score(F), and the number of features considered (#Features) for each technique.

Table 21 Result Summarization

	AC	PR	RC	F	#feat
XGBoost	0.960	0.933	0.939	0.936	21
XGBoost-strong feat_correl	0.948	0.904	0.936	0.919	14
XGBoost-univariate	0.962	0.923	0.955	0.938	17
XGBoost-RFE	0.958	0.919	0.938	0.928	17
XGBoost-L1 Reg	0.932	0.863	0.920	0.889	7
XGBoost-FI (thresh: 0.025)	0.960	0.929	0.945	0.937	15
XGBoost- FI (thresh: 0.013)	0.951	0.913	0.945	0.928	9

The following primary observations were made. Based on the evaluation metrics, the best values are obtained in Univariate feature selection, offering an increase of 0.21% in Accuracy, a decrease of 1.1% in Precision value, and 1.7% in the Recall score, 0.21% increase in F1-score. However, we could reduce the feature count by only 4. To determine if further feature reduction is possible, we relied on the feature_importance score, with a threshold of 0.025 and 0.013. The former threshold reduced the number to 9 (~57% decrease), resulting in almost the same accuracy, 95.1%, compared to training using all 21 features (96%). We could also achieve a higher recall (0.6%). However, for the later threshold value, we could reach a score of 96.2%, a slightly higher F1 score (93.6%), and a higher recall value (0.6%). Therefore, it is evident that not all 21 features are required for optimised model performance. Based on our need, we can opt for either of the two thresholds and train the model with as few as nine or a maximum of 15 features, yet attain comparable results when trained with all 21 features. Another advantage in our favour would be that the trained model would be less susceptible to overfitting.

Acknowledgement

The authors gratefully acknowledge the support and encouragement received from GITAM University in completing this work.

Conflicts of Interest

The author(s) declare(s) that there is no conflict of interest regarding the publication of this paper

References

[1] Mehbodniya, A., Lazar, A.J.P., Webber, J., Sharma, D.K., Jayagopalan, S., K, K., Singh, P., Rajan, R., Pandya, S. and Sengan, S., 2022. Fetal health classification from cardiotocographic data using machine learning. *Expert Systems,* 39(6): e12899. DOI: https://doi.org/10.1111/exsy.12899

[2] Rawat, T. and Khemchandani, V., 2017. Feature engineering (FE) tools and techniques for better classification performance. *International Journal of Innovative Engineering and Technology*. 8(2): 169-179. DOI: https://doi.org/10.21172/ijiet.82.024

[3] Karmarkar, A., Altay, A., Zaks, A., Polyzotis, N., Ramesh, A., Mathes, B., Vasudevan, G., Giannoumis, I., Wilkiewicz, J., Simsa, J. and Hong, J., 2020. Towards ML Engineering: A Brief History Of TensorFlow Extended (TFX). arXiv preprint arXiv:2010.02013. DOI: https://doi.org/10.48550/arXiv.2010.02013

[4] Uddin, M.F., Lee, J., Rizvi, S. and Hamada, S., 2018. Proposing enhanced feature engineering and a selection model for machine learning processes. *Applied Sciences,* 8(4): 646. DOI: https://doi.org/10.3390/app8040646

[5] Verdonck, T., Baesens, B. and Oskarsdottir, M., 2021. Special Issue on Advances in Feature Engineering editorial. *Machine Learning.* 113(7): 3917–3928. DOI: https://doi.org/10.1007/s10994-021-06042-2

[6] Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R.P., Tang, J. and Liu, H., 2017. Feature selection: A data perspective. *ACM computing surveys (CSUR),* 50(6): 1-45. DOI: https://doi.org/10.48550/arXiv.1601.07996

[7] Morán-Fernández, L. and Bolón-Canedo, V., 2021. Dimensionality Reduction: Is Feature Selection More Effective Than Random Selection?. In *International Work-Conference on Artificial Neural Networks.* 113-125. DOI: https://doi.org/10.1007/978-3-030-85030-2_10. Cham: Springer International Publishing.

[8] Ayres-de-Campos, D., Bernardes, J., Garrido, A., Marques-de-Sa, J.

and Pereira-Leite, L., 2000. SisPorto 2.0: a program for automated analysis of cardiotocograms. *Journal of Maternal-Fetal Medicine,* 9(5): 311-318. DOI: https://doi.org/10.1002/1520-6661(200009/10)9:5

[9] Sundar, C., Chitradevi, M. and Geetharamani, G., 2012. Classification of cardiotocogram data using neural network based machine learning technique. *International Journal of Computer Applications,* 47(14): 19–25. DOI: https://doi.org/10.5120/7256-0279

[10] Fasihi, M., Nadimi-Shahraki, M.H. and Jannesari, A., 2021. A shallow 1-D convolution neural network for fetal state assessment based on cardiotocogram. *SN Computer Science,* 2(4): 287. DOI: https://doi.org/10.1007/s42979-021-00694-6

[11] Garcia-Canadilla, P., Sanchez-Martinez, S., Crispi, F. and Bijnens, B., 2020. Machine learning in fetal cardiology: what to expect. *Fetal Diagnosis And Therapy*, 47(5): 363-372.DOI: https://doi.org/10.1159/000505021

[12] Mohannad, A., Shibata, C., Miyata, K., Imamura, T., Miyamoto, S., Fukunishi, H. and Kameda, H., 2021. Predicting high risk birth from real large-scale cardiotocographic data using multi-input convolutional neural networks. *Nonlinear Theory and its Applications, IEICE,* 12(3): 399-411. DOI: https://doi.org/10.1587/nolta.12.399

[13] Guyon, I. and Elisseeff, A., 2003. An introduction to variable and feature selection. *Journal Of Machine Learning Research,* 3(Mar): 1157-1182. DOI: https://doi.org/10.5555/944919.944968

[14] Sahin, H. and Subasi, A., 2015. Classification of the cardiotocogram data for anticipation of fetal risks using machine learning techniques. *Applied Soft Computing*, 33: 231-238. DOI: https://doi.org/10.1016/j.asoc.2015.04.038

[15] Pudjihartono, N., Fadason, T., Kempa-Liehr, A.W. and O'Sullivan, J.M., 2022. A review of feature selection methods for machine learning-based disease risk prediction. *Frontiers in Bioinformatics,* 2: 927312. DOI: 10.3389/fbinf.2022.927312

[16] Ng, A., Crowe, R., Moroney, L., & Arámburu, C. B. (2023). Machine learning engineering for production (MLOps) specialization [Online course]. Coursera. https://www.coursera.org/learn/introduction-to-machine-learning-in-production

[17] Heaton, J., 2016, March. An empirical analysis of feature engineering for predictive modeling. In Proceedings of the IEEE Southeast Conference 2016. 1-6. IEEE. DOI: 10.1109/SECON.2016.7506650.

[18] Awad, M. and Fraihat, S., 2023. Recursive feature elimination with cross-validation with decision tree: Feature selection method for machine learning-based intrusion detection systems. *Journal of Sensor and Actuator Networks,* 12(5): 67. DOI: https://doi.org/10.3390/jsan12050067

[19] Liu, M., Xu, C., Luo, Y., Xu, C., Wen, Y. and Tao, D., 2017. Cost-sensitive feature selection by optimizing F-measures. *IEEE Transactions on Image Processing*, 27(3): 1323-1335. DOI: https://doi.org/10.1109/tip.2017.2781298

[20] Abiyev, R., Idoko, J. B., Altıparmak, H., & Tüzünkan, M. 2023. Fetal health state detection using interval type-2 fuzzy neural networks. Diagnostics, 13(10): 1690. DOI: https://doi.org/10.3390/diagnostics13101690

[21] Jebadurai, I., Paulraj, G., Jebadurai, J., & Silas, S. 2022. Experimental analysis of filtering-based feature selection techniques for fetal health classification. *Serbian Journal of Electrical Engineering*, 19(2): 207–224. DOI: https://doi.org/10.2298/sjee2202207j

[22] Regmi, B., & Shah, C. 2024. *Classification Methods Based on Machine Learning for the Analysis of Fetal Health Data.* ArXiv.org. https://arxiv.org/abs/2311.10962