ALBUKHARY INTERNATIONAL UNIVERSITY

SCHOOL OF COMPUTING AND INFORMATICS

SEMESTER 2 2024/2025

CCS2213

Machine Learning
Individual Assignment

Assignment Title: Fetal Health Classification Using Support Vector Machine and Decision Tree

Lecturer: Associate Professor Dr. Umi Kalsom binti Yusof

NAME	ID	GROUP	DATASET
MD RASHIDUL ISLAM	AIU23102071	A	Fetal Health Classification (Kaggle)

Table of Contents

1.0 Dataset Background

1.1 Introduction

The data for the Fetal Health Classification dataset includes 2,126 tests called Cardiotocograms (CTGs). Obstetricians use these tests on pregnant women to observe the heart rate of the baby, the baby's movements and contractions of the uterus. Experienced doctors have reviewed and labelled all the data included in this dataset. A record can be grouped Normal, Suspect or Pathological.

The purpose of this dataset is to support the creation of tools that identify healthy or dangerously ill fetuses. A discovery of a problem early on gives doctors more time to respond. These interventions could save the lives of moms and their newborns where access to adequate health care is scarce (Kaggle, 2021).

1.2 Background Study (Literature Review)

The study shows that using machine learning helps classify the health of a fetus in CTG data. Yalamanchili et al. (2024) focused on Random Forest, SVM and Decision Trees methods and were able to reach 93% accuracy. Even so, they observed that working datasets need to be larger and more representative for applications in the real world. According to Yin and Bingi (2023), their models achieved 99.59% correctness through SVM and XGBoost. They also created a way to explain their decisions, so doctors can more easily understand the predictions. Best test results were obtained by Alam et al. (2022) when Random Forest was used with a high accuracy of 97.51%. They pointed out errors involving overfitting and a lack of balance between the numbers of samples, suggesting that better data strategies should be used.

1.3 Class Distribution

Normally healthy fetuses account for most of the data at 1,655 records, followed by 295 suspect records and 176 pathological records.

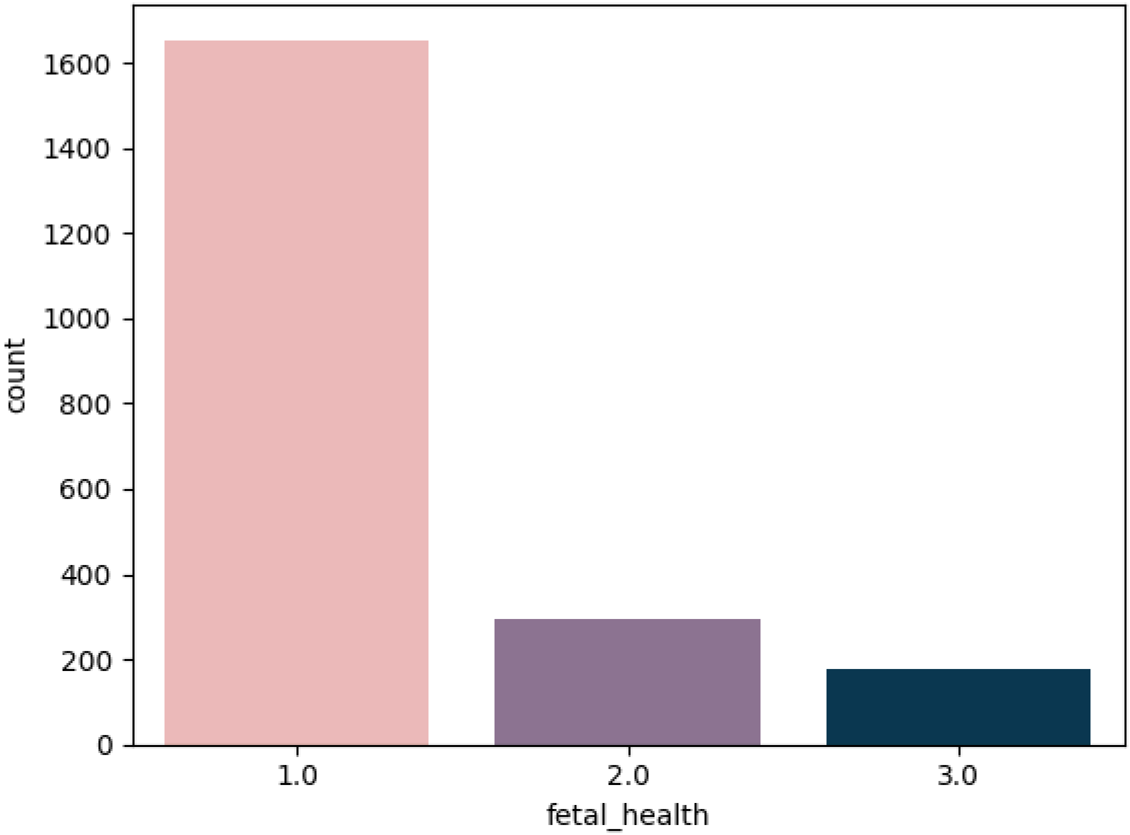


Figure 1: Original class distribution showing imbalance

It is clear from Figure 1 that there is an imbalance. There is so much Normal data that the model may learn to favor it. Often, it may predict “Normal” when it should detect “Suspect” or “Pathological” cases.

1.4 Data Balance Assessment

The dataset started out with a major minority in the Suspect and Pathological classes, whereas the majority of examples belonged to the Normal class (Figure 1). Consequently, the models might typically identify the majority class and overlook important cases. So, I used manual oversampling that copied samples from the smaller classes to reach a target of each class with 1,655 samples. After balancing (Figure 2), the data for each class were equal, which enabled the model to see all types of fetal conditions more clearly.

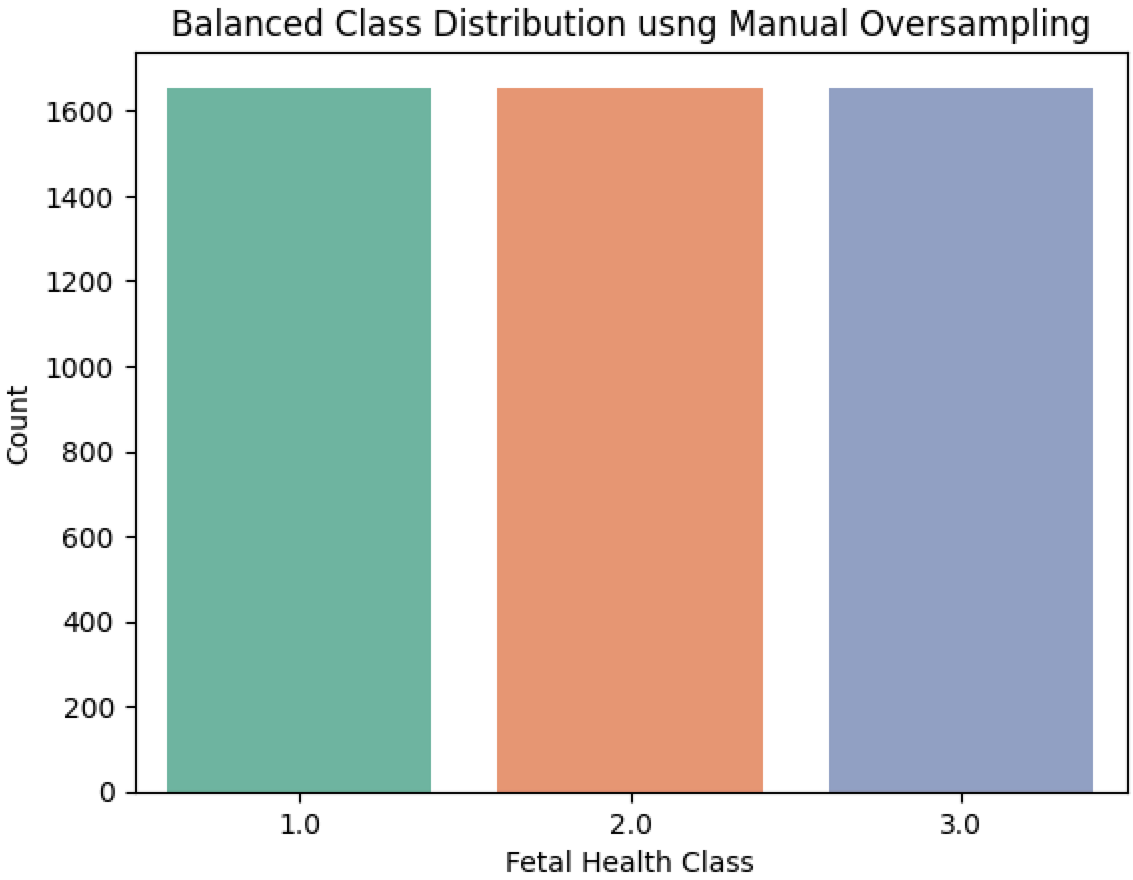


Figure 2: Balanced class distribution using manual oversampling

This data balancing step is crucial because it enhances the model's ability to detect risky cases and assists avoid biased results that could lead to missed medical warnings.

2.0 Pre-processing Options

2.1 Missing Values and Cleaning

There is no missing value in the dataset. We do not have to fix any data since Figure 3 shows that there are 2,126 entries for each column. Because all the features are numeric, machine learning models can use them directly.

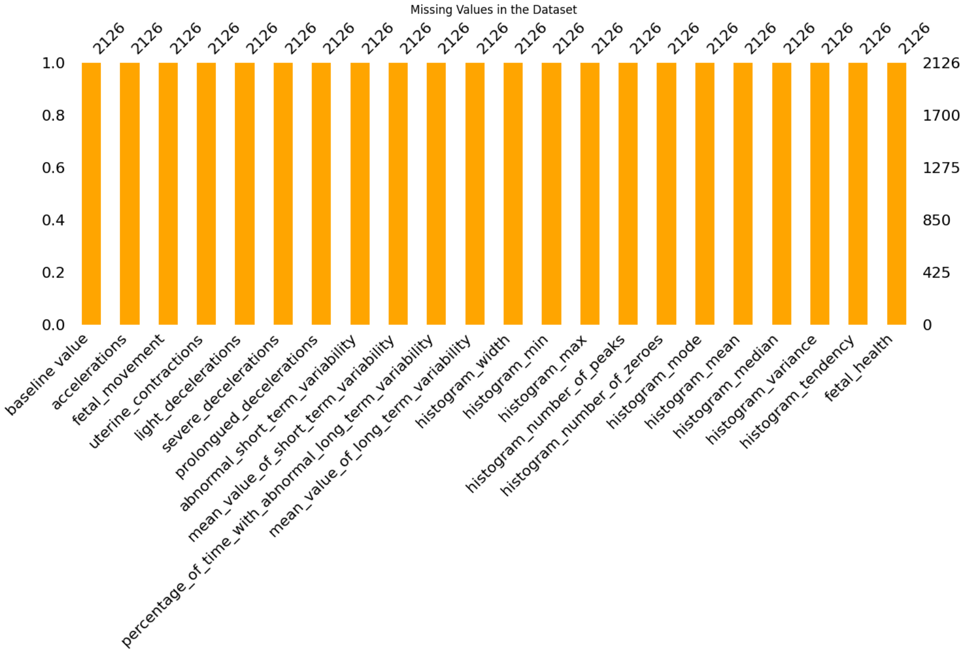


Figure 3: No missing values in the dataset.

2.2 Feature Scaling

Features have big numbers and others have small numbers. Such models as SVM are vulnerable to changes in scale. For this reason, I applied StandardScaler to all of the features. With this method, their mean becomes 0 and their standard deviation is 1.

2.3 Splitting the Dataset

I split the training and test cases in 80% and 20%, using a stratified approach. The model is able to perform well with every class because of this allocation.

3.0 Model Evaluation Technique

3.1 Choosing the Right Evaluation Method

I selected the hold-out method to test and evaluate the machine learning models this time. As a result, the dataset was separated into Training set (80%) and Test set (20%) for testing. To see how the model performs with unseen data ,set aside 20% for testing.

Both sets were created using a stratified technique, maintaining the same number of classes in each one. This is necessary since the initial dataset has is unbalanced. Using stratification allows all three groups (Normal, Suspect and Pathological) to be present in the training data and test data.

3.2 Why Hold-Out Was Suitable

The dataset contains 2,126 observations which is a moderate amount. For this reason, hold-out was selected as the evaluation strategy. Since it's easy and quick, it gives me a good sense of how well your model performs. The data was divided using stratification to ensure each class had a similar number of cases. I found that on large data like ours, using the hold-out method was more appropriate and gave equally accurate results.

3.3 Insights from the Evaluation

I collected scores of accuracy and macro F1-score to compare the results from the tested models (SVM and Decision Tree) against each other. Using RFE with the Decision Tree achieved an accuracy of 91.5%. By getting rid of unimportant features, we clearly made our classification stronger. However, these methods were effective for creating graphs, but did not raise the accuracy. It means that choosing the right features makes a bigger difference than just lowering the number of variables.

3.4 Contribution of the Evaluation

The evaluation showed that the Decision Tree with RFE method demonstrated the best results and highest accuracy. I found that removing unnecessary features increases the accuracy of the model. PCA and LDA might help with understanding the data through charts, though using them wrong can lessen the data's accuracy. All in all, the project reveals that selecting the correct evaluation technique and choosing which features to include helps improve fetal health prediction, even on a balanced and fairly sized dataset.

4.0 Choice of Classifier

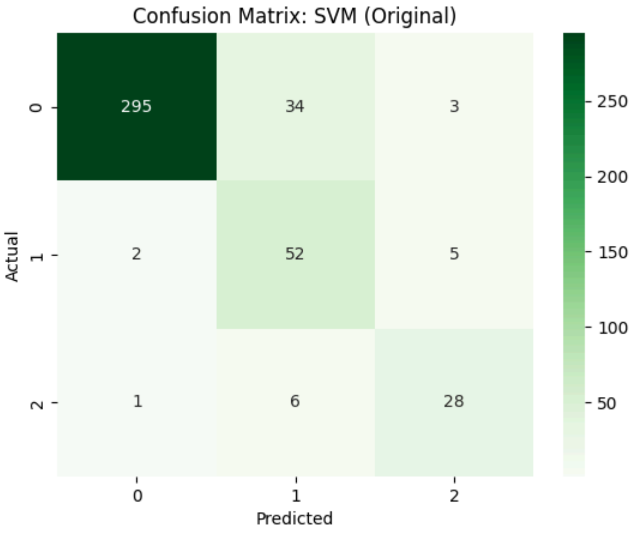
4.1 Selected Models: SVM and Decision Tree

The classifiers selected for this project were Support Vector Machine (SVM) and Decision Tree (DT). I picked these models since they serve well in classification, particularly for datasets that have a numeric format, like this one.

When the data consists of many numbers and is multidimensional, SVM does well. It clearly divides groups and is able to resolve tough problems such as predicting the health of babies. SVM achieved excellent accuracy in similar CTG experiments, proving it is reliable, as reported Yin & Bingi (2023). The explanation for Decision Trees is straightforward which helps medical staff make sense of the results. Alam et al. (2022) explain that DT does well with uneven data and is able to detect unusual relationships between the features.

4.2 Performance Comparison and Metrics

I achieved the accuracy and macro F1 score using the confusion matrix and classification report. When using SVM (Original), accuracy was 88.03% and the macro F1 score was 0.8047. The model performed strongly for the "Normal" class but had lower precision for the "Suspect" class which is shown in Figure 4. The original Decision Tree was slightly more accurate with an accuracy of 89.67% and a macro F1 score of 0.812. It gave better overall scores across all groups and this was particularly clear in the “Pathological” group (F1 score 0.83) which is shown in Figure 5.



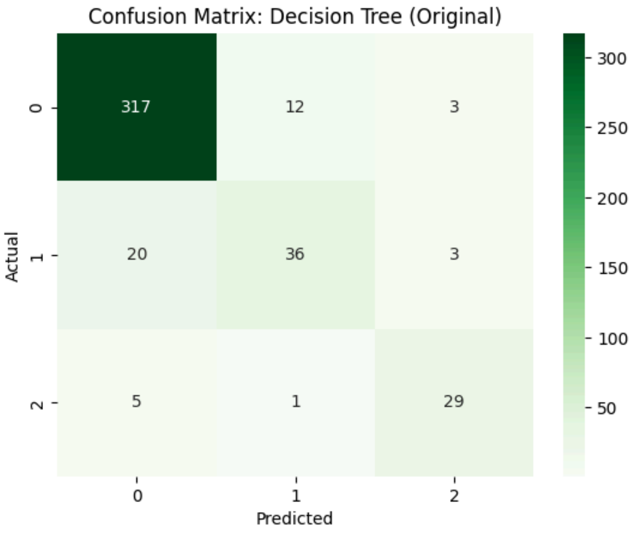


Figure 4: Confusion matrix SVM(Original)   *Figure 5: Confusion matrix DT(Original)*

The confusion matrix shows that while both models are strong, Decision Tree is more balanced, making fewer mistakes on the minority classes.

4.3 Chosen Evaluation Metric

Because the dataset is imbalanced, I selected the macro F1 score as the key evaluation score. Since Suspect and Pathological are represented by fewer samples, it is beneficial to give all classes the same importance. This method was also used to show which classes were wrongly classified most of the time.

These numbers give us a clearer understanding of how every model performs, apart from its general accuracy.

5.0 Dimensionality Reduction

5.1 Purpose and Justification

Dimensionality reduction makes the data easier to work with because it cuts down on the amount of information. As a result, models can be trained faster, are less likely to learn too much data and the results are easier to see. As our total features were 21, we used different techniques to reduce the data and observe their effect on model performance.

5.2 Feature Extraction

Feature extraction makes new features from the original ones, trying to keep most of the important information. The most common ways are to use PCA and LDA.

5.2.1 Principal Component Analysis (PCA)

PCA helps identify these direction by selecting those that hold the directions with the most variance. I used PCA and captured 95% of the variance, the number of features was reduced from 21 to 14.

Even so, the results in the confusion matrices suggest a small decrease in accuracy for SVM + PCA: 87.1% and for DT + PCA: 87.2%

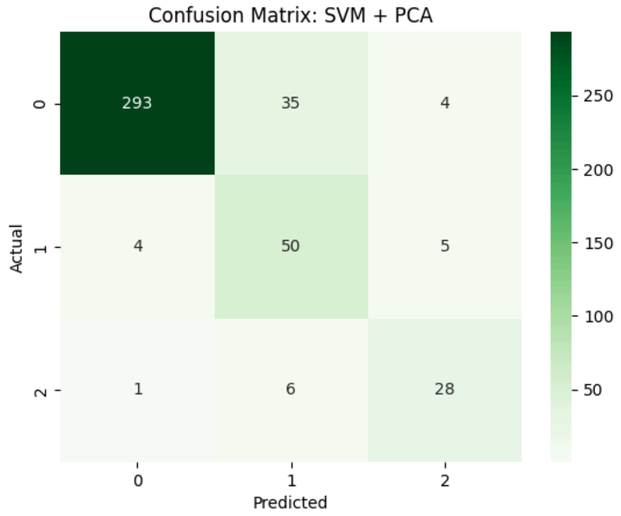


Figure 6: Confusion matrix SVM + PCM

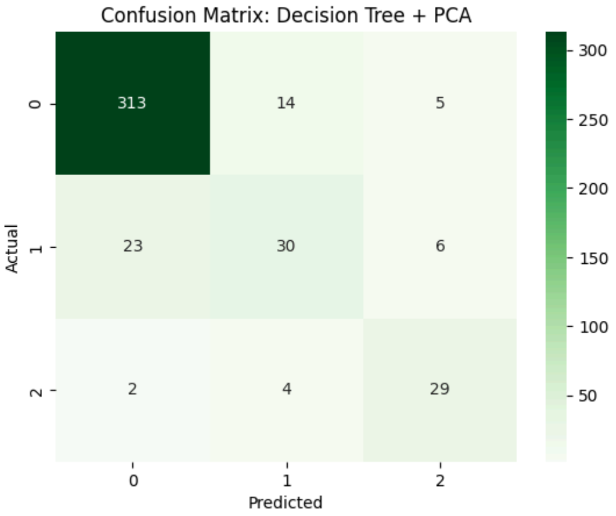


Figure 7: Confusion matrix DT + PCM

It was caused because class labels are not used when converting with PCA. It might keep characteristics that differ a lot but do not help with classification which can decrease the machine's accuracy.

5.2.2 Linear Discriminant Analysis (LDA)

LDA reduces the number of dimensions by attending to features that distinguish different classes the best. Features were cut down to 2 components to fit with 3 classes. Even with new training, the result remained the same. In reality, the accuracy dropped: SVM + LDA: 83.6% and DT + LDA: 86.6%.

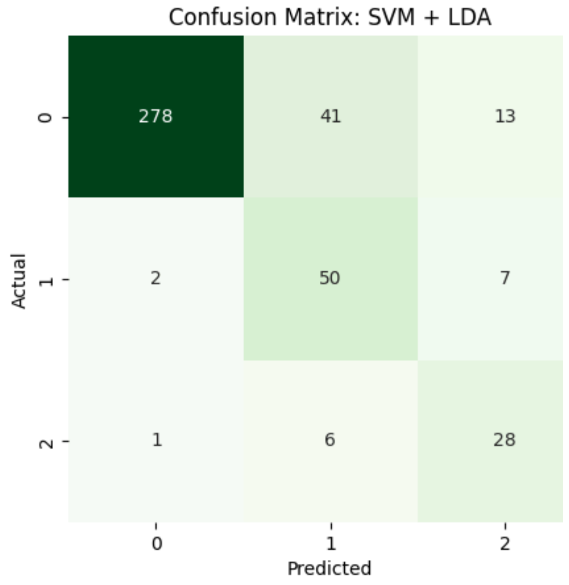


Figure 8: Confusion matrix SVM+LDA

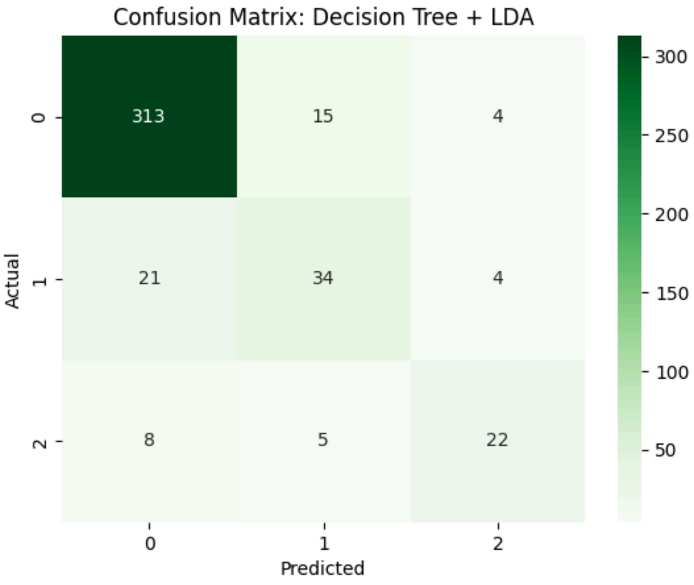


Figure 9: Confusion matrix DT+LDA

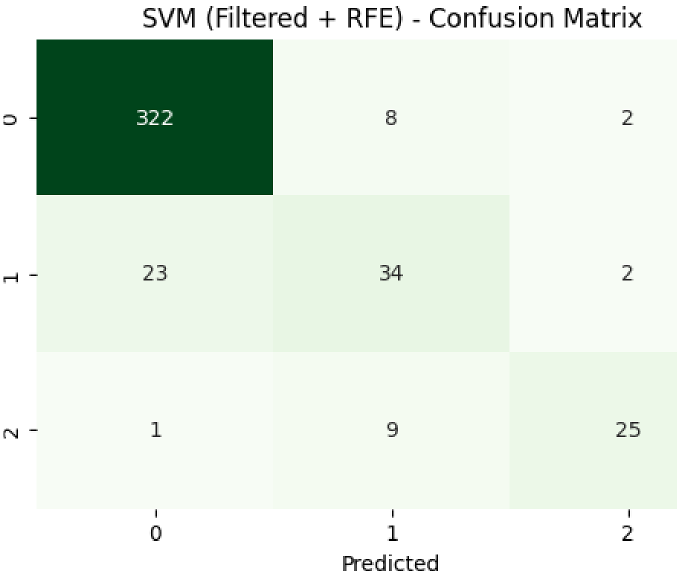
Although LDA helps with visualisation, reducing the features too much can cause loss of key information, leading to lower accuracy.

5.3 Feature Selection

Instead of making extra features, feature selection decides on the best available ones. Our approach combines a filter method called correlation with a wrapper approach called Recursive Feature Elimination or RFE.

5.3.1 Filter and Wrapper Methods

Applying the filter method, the least relevant features were removed so as to not include irrelevant data. Then, the wrapper method RFE together with the SVM estimator was applied to decide on the top 10 most significant features. Choosing features in two steps made the model perform much better. As it is shown in the confusion matrices, Decision Tree (Filtered + RFE) got an accuracy of 91.5%. Only SVM (Filtered + RFE) scored the same as or higher than the Decision Tree model (Filtered + RFE). Because unnecessary factors no longer caught their attention, the models concentrated better on the data that mattered for classification.



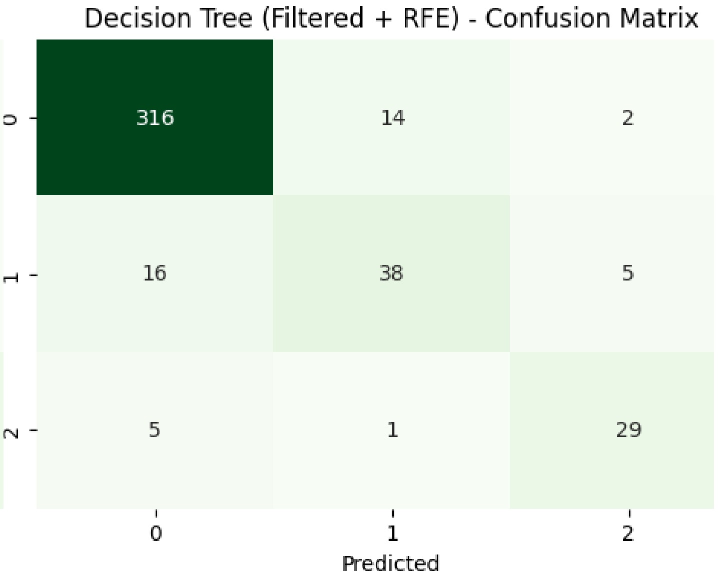


Figure 10: Confusion matrix SVM Filtered +RFE   Figure 11: Confusion matrix DT

5.4 Comparative Analysis

The examination of the eight model combinations reveals how each technique affects the accuracy of the forecasts. Figure 12 illustrates that the combination of a Decision Tree and Filtering + Recursive Feature Elimination led to 91.5% accuracy, whereas SVM with LDA came in lowest at 83.6%. It is obvious that the performance is strongly influenced by the selected dimensionality reduction or selection method. Tequila detection was successful for models with no feature changes, as seen in 88.0% for SVM and 89.7% for Decision Tree.

However, the results from both PCA and LDA were less accurate than those obtained directly from the data. Loss of important information usually occurs for multiclass problems such as fetal health classification, as these methods usually reduce the number of feature dimensions. SVM combined with PCA performed best at 87.1%, but using LDA the accuracy fell to 83.6%. These techniques are generally helpful for visualizing large datasets or when the data is very big, although, in this case, they didn't work and may actually have removed elements that helped the dataset be separated into classes.

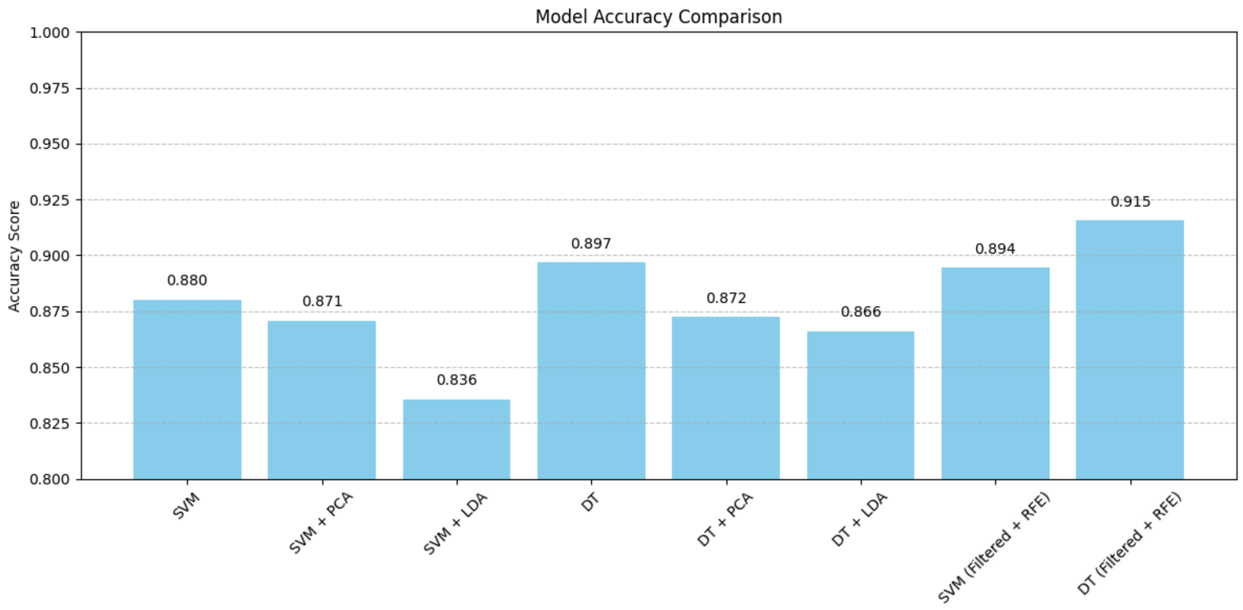


Figure 12: Model accuarcy comparison

Alternatively, using Filtered + RFE techniques significantly enhanced the performance. This approach concentrated on finding the most important features which helped to remove the noise and prevented overfitting. SVM backed by Filtered and Recursive Feature Elimination performed the best at 89.4%, but Decision Tree with Filtered + RFE outperformed every other method. This confirms that selecting appropriate features plays a larger role than design simplification. Choosing a set of important features made the models more accurate and dependable.

6.0 Conclusion

This project utilised machine learning to predict fetal health conditions based on CTG data. After oversampling the classes with high and low risk, the models showed improved results according to each fetal health category. Using these preprocessing techniques such as scaling and stratified splitting, improved the accuracy of the results. Using RFE, Decision Tree achieved a high score with an accuracy of 91.5%. Selecting features worked better for this problem than reducing the number of dimensions, as PCA and LDA had some impact on the performance. These results indicate that holding onto key features is preferable to just cutting the total number of them.

Overall, balancing the data, selecting the needed features and using proper assessment created a powerful model for predicting fetal health. Following these steps can help doctors guarantee safer pregnancies.

References

Alam, M. T., Khan, M. A. I., Dola, N. N., Tazin, T., Khan, M. M., Albraikan, A. A., & Almalki, F. A. (2022). Comparative analysis of different efficient machine learning methods for fetal health classification. *Applied Bionics and Biomechanics*, Article ID 6321884. https://doi.org/10.1155/2022/6321884

Kaggle. (2021). *Fetal health classification dataset*. https://www.kaggle.com/datasets/andrewmvd/fetal-health-classification

Yalamanchili, S., Mohanty, S. N., Ramesh, J. V. N., Yang, M., & Chalapathi, M. M. V. (2024). Cardiotocography data analysis for fetal health classification using machine learning models. *IEEE Access, 12*, 26005–26010. https://doi.org/10.1109/ACCESS.2024.3364755

Yin, Y., & Bingi, Y. (2023). Using machine learning to classify human fetal health and analyze feature importance. *Biomedinformatics, 3*(2), 280–298. https://doi.org/10.3390/biomedinformatics3020019