Academic Report: Analysis and Classification of Fetal State using Cardiotocography Data

**Abstract**

This report details the process of analyzing and classifying fetal state using cardiotocography (CTG) data from the "CTG.xls" dataset. The study involved data loading, cleaning, preprocessing, exploratory data analysis (EDA), and the development and evaluation of machine learning models to predict fetal state (Normal, Suspect, or Pathologic). Key steps included handling header/footer rows, setting correct column names, addressing missing values and outliers, scaling numerical features, and visualizing data distributions and relationships. Three classification models—Logistic Regression, Decision Tree, and Random Forest—were trained and evaluated. The Random Forest model, both in its default and hyperparameter-tuned versions, demonstrated superior performance based on accuracy and AUC score, indicating its potential for accurate fetal state prediction.

**1. Introduction**

Cardiotocography (CTG) is a widely used non-invasive technique for monitoring fetal heart rate and uterine contractions during pregnancy and labor. Analysis of CTG data can provide valuable insights into fetal well-being and help identify potential distress. This study aims to leverage machine learning techniques to automatically classify fetal state based on CTG features, providing a potential tool to assist clinicians in their assessment.

**2. Data Description and Loading**

The dataset used in this study is sourced from the "CTG.xls" file, specifically the "Raw Data" sheet. The dataset contains various features extracted from CTG recordings, along with a classification of fetal state. The header descriptions were used to understand the meaning of each column. The data was loaded into a pandas DataFrame for subsequent processing.

**3. Data Cleaning and Preprocessing**

Initial inspection of the data revealed the presence of extraneous header and footer rows, which were removed to ensure that only relevant data was included in the analysis. The correct header row, containing descriptive column names, was identified and set as the DataFrame's columns. The original header row was then removed from the data.

Missing value analysis showed no missing values in the dataset, eliminating the need for imputation or removal of rows/columns based on missingness.

Column names were standardized to be more descriptive and consistent with the header descriptions, improving the readability and interpretability of the data.

Data types were converted to ensure appropriate handling of numerical and categorical features. Numerical columns were converted to a numeric format, with errors coerced to NaN. The 'Date' column was converted to a datetime format, while identifier columns ('FileName', 'SegFile') were kept as object type.

Outliers in numerical columns were addressed using the Interquartile Range (IQR) method. Values falling outside 1.5 times the IQR were identified and replaced with the median of their respective columns to mitigate their potential impact on model performance.

**4. Exploratory Data Analysis**

Exploratory data analysis was conducted to understand the characteristics of the dataset and the relationships between features.

* **Column Type Identification:** Columns were categorized into numerical and categorical types. The dataset was found to contain 37 numerical columns and no categorical columns (excluding identifier and date columns).
* **Numerical Column Visualization:** Histograms and boxplots were generated for all numerical columns to visualize their distributions and identify potential skewness or outliers. Many columns exhibited skewed distributions, and while outlier handling was performed, some columns still showed values outside the typical range in boxplots.
* **Correlation Analysis:** A correlation heatmap was created to visualize the pairwise correlations between numerical columns. The heatmap revealed varying degrees of correlation, with some features showing strong positive or negative relationships. This analysis helps understand feature dependencies and can inform feature selection or engineering efforts.

**5. Machine Learning Modeling**

The cleaned and preprocessed data was used to build and evaluate machine learning models for predicting fetal state.

* **Data Preparation:** The data was prepared for modeling by separating features (X) from the target variable (y), which is 'NSP (fetal\_state)'. Identifier columns ('FileName', 'Date', 'SegFile') were excluded from the feature set.
* **Data Splitting:** The dataset was split into a training set (80%) and a testing set (20%) to train and evaluate the models, respectively. This ensures that the models are evaluated on unseen data, providing a more realistic estimate of their performance.
* **Model Selection:** Three classification models were selected for this task: Logistic Regression, Decision Tree Classifier, and Random Forest Classifier. These models were chosen for their varying approaches to classification and their common use in similar problems.
* **Model Training:** Each selected model was trained on the training data.
* **Model Evaluation:** The performance of each trained model was evaluated on the testing data using several metrics: accuracy, classification report (precision, recall, F1-score), confusion matrix, and AUC score. The evaluation metrics provide a comprehensive understanding of each model's ability to correctly classify fetal states, including its performance on individual classes (Normal, Suspect, Pathologic).
* **Hyperparameter Tuning:** Hyperparameter tuning was performed for the Random Forest model using GridSearchCV to find the optimal combination of parameters that maximizes performance.

**6. Results and Discussion**

The evaluation of the trained models revealed that all three models achieved good performance in classifying fetal state. The Random Forest model, both in its default and tuned versions, consistently showed the highest accuracy and AUC scores.

* **Logistic Regression:** Achieved a respectable accuracy and AUC, demonstrating the effectiveness of a linear model for this problem.
* **Decision Tree:** Showed high accuracy, highlighting its ability to capture complex relationships in the data.
* **Random Forest (Default):** Outperformed Logistic Regression and the single Decision Tree in terms of AUC, indicating better overall discriminatory power.
* **Random Forest (Tuned):** Showed a marginal improvement in AUC compared to the default Random Forest, confirming the benefit of hyperparameter optimization.

The high accuracy and AUC scores, particularly for the Random Forest models, suggest that these models have strong potential for accurately predicting fetal state based on CTG features. However, considering the class imbalance in the dataset, it is crucial to examine the classification reports to understand the models' performance on the minority classes ('Suspect' and 'Pathologic'). High recall for these classes is particularly important in a clinical setting to minimize false negatives.

**7. Conclusion**

This study successfully analyzed and classified fetal state using CTG data. The data cleaning, preprocessing, and EDA steps were crucial in preparing the data for modeling. The evaluated machine learning models, especially the Random Forest, demonstrated promising results in predicting fetal state. While the high accuracy is encouraging, further clinical validation and consideration of the models' performance on minority classes are essential before potential deployment in a healthcare setting. Future work could explore other advanced machine learning techniques, feature engineering, and strategies specifically designed to handle imbalanced datasets to further improve classification performance.

**References**

* CTG.xls dataset (Source: specify source if available)
* Header descriptions (Source: Booklet)
* Scikit-learn documentation (for machine learning models and metrics)
* Pandas documentation (for data manipulation)
* Matplotlib and Seaborn documentation (for data visualization)