ANEMIA LEVEL PREDICTION IN CHILDREN CASE_STUDY

Utilizing Predictive Analytics for Anemia severity in children.

COLLABORATORS

1. Ruth Kitasi
2. Agatha Nyambati
3. Joseline Apiyo
4. Cecilia Ngunjiri
5. John Mbego
6. Leonard Koyio

PROJECT SUBMISSION DATE

22nd November 2024

1. Data Loading and Initial Exploration

The first step in any data science project is to understand the data. For this project, we began by loading the provided dataset, which contained various features related to socio-economic and health-related attributes of children, with the goal of predicting anemia levels.

We used standard data manipulation libraries to read the dataset into a structured format (pandas Data Frame), which allowed us to easily explore its contents. Upon loading the dataset, we conducted an initial exploration to understand its structure and quality. This involved:

- Checking the number of rows and columns in the dataset. Which is 33,924 rows and 17 columns
- Identifying the types of each feature. The categorical data is denoted by the data type 'object' and the numerical data is denoted by the data type 'float64' and 'int64'
- Identifying missing or null values across the dataset.
- Summarizing basic statistics for numerical variables, including measures like mean, median, 25th percentile, 50th percentile, 75th percentile and standard deviation.

This initial inspection helped us identify potential issues like missing values, which were critical in determining the next steps for data cleaning and transformation.

2. Data Cleaning and Preprocessing

After the initial exploration, we moved on to cleaning and preprocessing the data to prepare it for modeling. We identified several issues that needed attention:

- Missing Values: A significant challenge in our dataset was the presence of missing values, especially in the target variable (anemia level). To handle these missing values, we employed different strategies:
  - For missing values in the target variable (anemia level), we decided to remove rows where the target variable was missing, as imputing them could distort the classification process.
  - For categorical features with missing values, we imputed the missing values using ‘Unknown’.
  - For numerical features, we dropped rows with missing values to avoid introducing any uncertainty or errors that could arise from imputation methods.
  - Dropped unnecessary columns
  - Renaming Columns. We renamed several columns for clarity and ease of interpretation

ENCODING

We prepared our dataset for analysis by encoding categorical columns and then combining them with numerical columns. Here’s a breakdown of the steps we took:

Encoding Categorical Columns: We used ordinal encoding to convert categorical variables, which have a meaningful order, into numerical format. This allows them to be utilized in machine learning models. After encoding, we transformed the result back into a Data Frame to maintain the original column names.

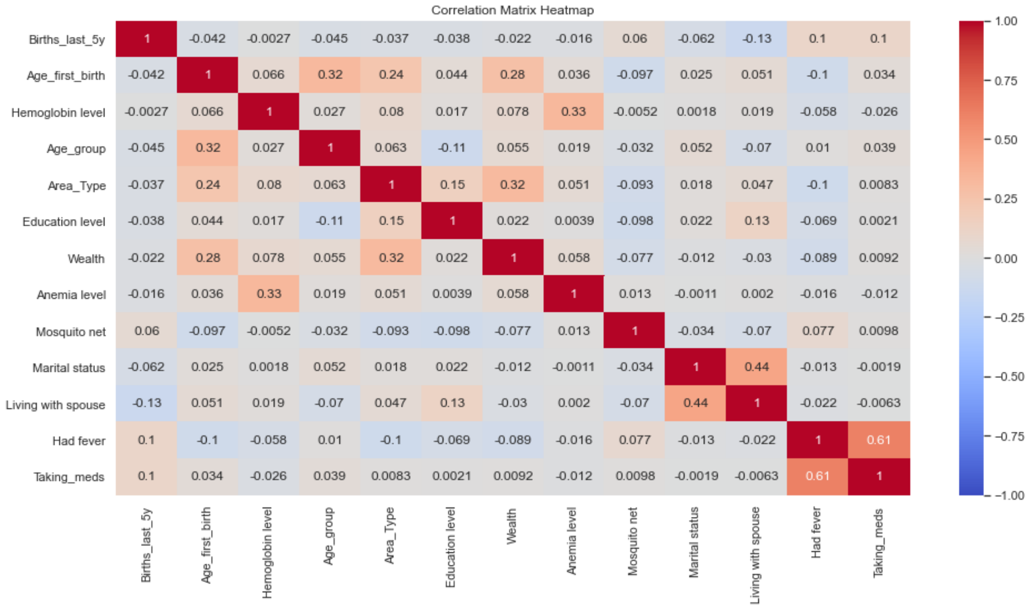
Data Type Conversion: We converted the encoded categorical columns from float to integer type, ensuring that the data is in the correct format for analysis.

Concatenating Datasets: We checked the shapes of our numerical and encoded categorical datasets to confirm they had the same number of rows. Then, we reset the indices of both datasets to ensure alignment before concatenating them into a single Data Frame.

Final Data Frame: We created a new Data Frame that combines both the numerical and categorical data, resulting in a comprehensive dataset ready for further analysis.

3. EDA

Heat Map



The heat map showed the correlations between various features, highlighting that factors like wealth, education, and marital status are linked to anemia levels, fever occurrence, and medication use. Strong correlations were found between "Had fever" and "Taking meds" (0.61), and "Living with spouse" has moderate correlations with anemia and fever. Overall, the analysis helped reveal how socio-economic and health factors are interconnected.

4. Data Preprocessing

Data Splitting:

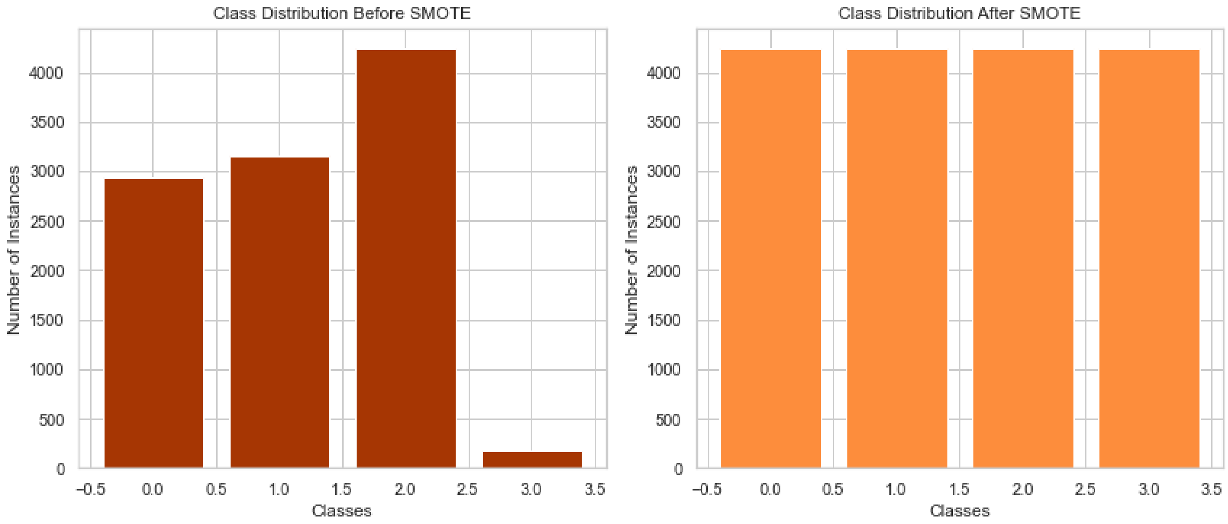
We divided the dataset into training and testing sets using an 80-20 split. The training set contains 10,508 samples, while the testing set has 2,628 samples. This step ensures that we can train our model on one subset and evaluate it on another, unseen set to assess how well it generalizes.

Feature Selection:

We selected relevant features by calculating the correlation matrix between the features and the target variable (Anemia level). Based on the correlations, we identified the top 10 features that are most strongly related to anemia, with Hemoglobin level showing the highest correlation (0.335). These features will be used for model training to improve performance and avoid overfitting.

Handling Class Imbalance:

We addressed class imbalance by applying SMOTE (Synthetic Minority Over-Sampling Technique), which oversamples the minority class to balance the distribution. Before SMOTE, there was a significant class imbalance, particularly in the "severe anemia" class. After applying SMOTE, the classes were evenly distributed, with each class containing 4,236 instances. This helps prevent bias in the model towards the majority class.



Standard Scaling:

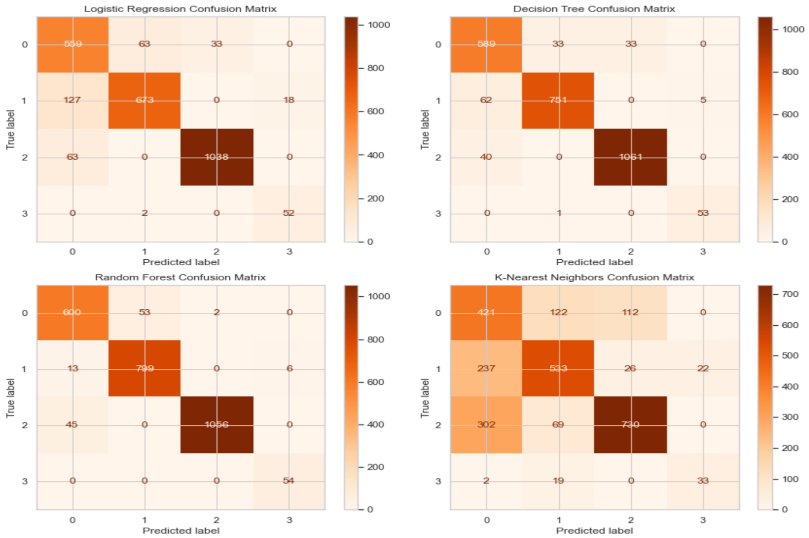
To standardize the features, we applied Standard Scaling to normalize the data, ensuring each feature has a mean of 0 and a standard deviation of 1. This step ensures that features with different ranges do not disproportionately influence the model's learning. We applied this scaling to both the resampled training data and the test set to ensure consistency during model evaluation.

These steps prepare the data for training a machine learning model that can predict anemia levels more effectively, with improved generalization and balanced class distribution.

5. Model Building and Evaluation

- Classifiers: We defined four classifiers: Logistic Regression, Decision Tree, Random Forest, and K-Nearest Neighbors (KNN).
- Pipeline: A machine learning pipeline was created for each model, where the classifiers were trained on resampled, scaled data (X_resampled_scaled, y_resampled) and evaluated on the test set (X_test_scaled).
- Performance Metrics: After training each model, we calculated key metrics—accuracy, precision, recall, and F1 score—and displayed them for comparison.
- Confusion Matrices: We visualized the performance of each classifier using confusion matrices to better understand the model predictions.

confusion matrix image



Logistic Regression Confusion Matrix

Decision Tree Confusion Matrix

Random Forest Confusion Matrix

K-Nearest Neighbors Confusion Matrix

Model Performance Comparison

- Random Forest consistently outperformed the other classifiers across all metrics (accuracy, precision, recall, F1 score), making it the top choice.
- KNN showed the weakest performance across all metrics, indicating it was less suitable for this dataset.

Hyper parameter Tuning with GridSearchCV

- GridSearchCV was used to fine-tune the Random Forest model by searching through a grid of hyperparameters (n_estimators, max_depth, etc.) to find the best combination for model performance.

- Best Hyper parameters: The grid search identified the optimal hyper parameters for Random Forest, which were then used to retrain the model.

- The tuned Random Forest model was evaluated on the test set, showing a slight improvement in performance with an accuracy of 95.62% and high precision, recall, and F1 scores, indicating a well-balanced model.

6. Conclusion

The project involved several critical steps from data loading and cleaning to feature engineering, model selection, and evaluation. Key aspects of the project included handling missing data, addressing class imbalance with SMOTE, and fine-tuning the model through hyperparameter optimization. The Random Forest model proved to be the best performer, demonstrating strong accuracy, precision, recall, and F1 scores.

Given its performance, the Random Forest model could serve as a reliable tool for predicting childhood anemia based on socio-economic and health-related factors, with the potential for real-world application in healthcare settings. Further validation in real-world contexts is recommended to assess its generalizability and robustness.