Challenges Faced During Data Analysis and Model Development

1. Class Imbalance

- ### Challenge:

  The dataset was highly imbalanced, with a significantly higher proportion of non-donors (Class 0) compared to donors (Class 1). This imbalance caused machine learning models to be biased toward the majority class, leading to:

- Higher accuracy but poor F1-scores for Class 1 (donors).
- Misleading evaluation metrics that did not reflect the model's ability to predict the minority class accurately.

- ### Technique Used:

- Oversampling of the minority class using techniques such as SMOTE (Synthetic Minority Oversampling Technique).
- This method generates synthetic examples of the minority class, ensuring the dataset is more balanced and improving the model's ability to learn patterns for Class 1.

- ### Reason:

  Balancing the dataset ensures the models give fair attention to both classes, improving F1-scores for donors and making the predictions more reliable for practical use.

2. Feature Scaling

- ### Challenge:

  The dataset features (e.g., "Months since last donation", "Number of Donations", "Months since first donation") had different scales. For instance:

- " Months since last donation " values ranged in months, while "Total Volume Donated " values were measured in c.c.

- Models that rely on distance metrics, such as KNN or Support Vector Classifier (SVC), were disproportionately influenced by features with larger scales, leading to suboptimal performance.

- **Technique Used:**
- Min-Max Normalization was applied to scale all features to a consistent range (e.g., [0, 1]).

- **Reason:**
- Normalization ensures all features contribute equally during model training, improving the performance of models sensitive to feature magnitude.

3. Limited Feature Set

- **Challenge:**

  The dataset had only four features, which might not capture the full complexity of donor behaviour. The limited feature set restricted the models' ability to identify nuanced patterns and relationships.

- **Technique Used:**

- Focused on feature engineering by ensuring proper scaling and leveraging the given features effectively.
- Considered simplicity and interpretability as priorities for model selection.

- **Reason**:

  While additional features could enhance model performance, the focus was on creating a practical model using the given attributes, ensuring it could be deployed with minimal data requirements.

4. Model Optimization

- ### Challenge:

  Default hyperparameters often resulted in suboptimal performance, as seen in initial model evaluations. For example:

- Random Forest without tuning showed lower accuracy compared to its potential after optimization.
- Logistic Regression parameters needed fine-tuning for regularization.

- ### **Technique Used**:

- Performed Grid Search for hyperparameter tuning on the top-performing models (Logistic Regression, Random Forest, and Naive Bayes).
- Explored parameters such as:

1. Logistic Regression: Regularization strength (C) and solver type.
2. Random Forest: Number of estimators, maximum tree depth, and minimum samples split.

- ### Reason:

  Hyperparameter optimization ensures the models operate at their full potential, improving accuracy and reliability for predictions.

5. Performance Evaluation Metrics

- ### **Challenge**:

  Accuracy alone was insufficient to evaluate model performance due to class imbalance. For instance:

- Models like Decision Tree and Random Forest had decent accuracy but poor F1-scores for Class 1.
- Support Vector Classifier (SVC) achieved high accuracy but failed completely to predict Class 1 (F1-score = 0.0).

- **Technique Used**:

- Focused on F1-scores for both classes, especially Class 1, to assess the model's ability to handle the minority class.
- Used a combination of metrics (Accuracy, F1-score) for comprehensive evaluation.

- **Reason**:

  Evaluation metrics must reflect the practical goal of accurately predicting donors, not just overall accuracy.

6. Model Deployment Preparation

- **Challenge**:

  Ensuring the final model was prepared for real-world deployment required:

- Saving the trained model for reuse.
- Keeping the implementation lightweight and interpretable.

- **Technique Used**:

- Saved the final Logistic Regression model as a Pickle file for easy integration into production systems.

- **Reason**:

  Pickle files allow the trained model to be reused without retraining, simplifying deployment and scaling for blood donation predictions.

Conclusion of Challenges and Techniques

Addressing these challenges ensured the development of a robust, reliable, and practical predictive model. The systematic approach to preprocessing, balancing, and optimizing models led to the selection of Logistic Regression, which achieved the highest accuracy (79.19%) and balanced F1-scores for both classes.