ICS5110 Final Report V1

2. Background

The project began with an idea to predict the type of weapon used in incidents based on a range of demographic and contextual features. Initially, the weapon categories included multiple classes such as "Firearm," "Poison," "Blunt Object," and others. The predictors encompassed parameters like "Victim Age," "Perpetrator Gender," and various relational and geographic attributes. The main goal was to develop a robust and efficient model capable of accurately classifying the weapon category.

Several machine learning algorithms were evaluated for this task, starting with the **Random Forest Classifier**, **XGBoost**, and **CatBoost**. While XGBoost and CatBoost offered faster execution times, Random Forest consistently produced the best results in terms of accuracy and balanced performance across categories. However, its longer training time was a challenge, taking several minutes to complete the process. To address this, the trained model was saved to a file, allowing it to be reused instantly in tools like **Gradio** for real-time predictions.

To ensure the **Random Forest** model could handle the categorical data effectively, **OneHotEncoding** was applied to all categorical features. This transformation converted each category into a set of binary features, making the data compatible with the Random Forest algorithm. Features like `State`, `Victim Sex`, and `Relationship Category` were encoded to ensure consistent and interpretable preprocessing.

Further performance improvements were achieved by grouping certain categorical features into broader categories. For example, months were combined into **seasons**, states were grouped into **regions**, and specific relationships like "Brother" and "Mother" were consolidated under the broader category of **Family**. These transformations simplified the dataset while retaining key patterns, improving both computational efficiency and model accuracy.

Despite these improvements, the dataset's inherent class imbalance emerged as a critical issue. For instance, "Firearm" records significantly outnumbered others like "Poison," which skewed the model’s ability to generalize across classes. To address this, weapon categories were consolidated into binary outputs: **Firearm** and **Non-Firearm**. While this reduced the granularity of predictions, it allowed for more balanced and meaningful outputs.

However, even after binarization, there was still an imbalance—approximately 80,000 records for "Firearm" compared to approximately 40,000 for "Non-Firearm." To address this, the **SMOTE (Synthetic Minority Oversampling Technique)** was applied. SMOTE generates synthetic examples of the minority class by interpolating between existing samples, thus creating a balanced dataset. This not only improved the model's recall for "Non-Firearm" but also ensured fairer performance across both classes.

2.1 Mechanics of Selected Machine Learning Techniques

- **Random Forest**: Builds multiple decision trees during training and combines their outputs to improve prediction accuracy. It mitigates overfitting by averaging predictions and supports both categorical and numerical data. For this project,

OneHotEncoding was applied to categorical features to ensure compatibility.

- **XGBoost**: A boosting algorithm that sequentially builds weak learners to correct errors from previous iterations. While it can handle categorical features natively, we maintained **OneHotEncoding** for consistency with other models.

- **CatBoost**: A boosting algorithm optimized for handling categorical data natively. Unlike Random Forest and XGBoost, it did not require **OneHotEncoding**, simplifying the preprocessing pipeline.

2.2 Rescaling and Normalization

While rescaling and normalization were not explicitly applied in this project due to the categorical nature of the majority of features, their importance lies in standardizing feature ranges. For models that rely on distance-based computations (e.g., SVM, KNN), rescaling ensures that features contribute equally to the model.

2.3 Cross-Validation

Cross-validation divides the dataset into multiple subsets (folds), ensuring the model is trained on different data splits for more reliable performance evaluation. While explicit cross-validation was not implemented, the **stratified train-test split** ensured that the class distribution in training and testing sets was consistent, reducing potential biases.

2.4 Dimensionality Reduction and Feature Selection

Feature selection was prioritized over dimensionality reduction in this project. By identifying and retaining the most relevant features, the dataset was simplified, improving interpretability, and reducing overfitting. Dimensionality reduction techniques like PCA were not used, as the dataset was structured and interpretable without further decomposition.

2.5 Quantitative Measurements

The following metrics were used to evaluate the models:

- **Accuracy**: Percentage of correct predictions.
- **Precision**: Proportion of true positives among predicted positives.
- **Recall**: Proportion of true positives among actual positives.
- **F1-Score**: Harmonic mean of precision and recall, balancing their trade-offs.
- **Support**: Number of instances for each class.

4. Experiments

4.1 Experiments Conducted

The experiments explored multiple algorithms and preprocessing techniques:

1. **Initial Model Selection**:
    - Random Forest, XGBoost, and CatBoost were tested for multi-class classification to identify the best-performing algorithm.
    - Random Forest demonstrated superior accuracy and balanced performance, while XGBoost and CatBoost were faster but struggled with imbalanced classes.
2. **Feature Engineering**:
    - Grouped features like months into **seasons** and states into **regions** to simplify the dataset without losing critical information.
    - Consolidated specific relationships (e.g., "Brother," "Mother") under broader categories like **Family** to improve computational efficiency.
3. **Class Consolidation**:
    - Converted weapon categories into binary labels: **Firearm** and **Non-Firearm**. This adjustment addressed the inherent complexity of multi-class classification and focused the predictions on the more significant dichotomy.
4. **Data Balancing**:
    - Applied **SMOTE (Synthetic Minority Oversampling Technique)** to address class imbalance. This ensured that the minority class, "Non-Firearm," was better represented in the training data, significantly improving recall for this class.
5. **Encoding Strategy**:
    - Used **OneHotEncoding** to preprocess categorical features (e.g., `State`, `Victim Sex`, `Relationship Category`) to ensure compatibility with algorithms like Random Forest and XGBoost.
6. **Model Persistence**:
    - Experimented with saving trained models to files (e.g., `.pkl` format) after training. This optimization allowed tools like **Gradio** to query the saved model directly, enabling instantaneous predictions without rerunning the training phase.
7. **Data Splitting**:
    - Tested different data splits for training and testing (80/20 and 70/30) to analyze the impact on model performance. The standard 80/20 split provided the best balance of training efficiency and testing reliability.

4.2 Implementation of Machine Learning Techniques

- **Random Forest**: Used with `class_weight='balanced'` and required **OneHotEncoding** for categorical data compatibility.
- **XGBoost**: Leveraged **OneHotEncoded** features with gradient boosting for faster training.
- **CatBoost**: Natively handled categorical data, eliminating the need for **OneHotEncoding** but showed lower precision compared to Random Forest.

4.3 Results and Comparisons

4.3.1Algorithms compared (initial experiments)

Model	Precision (Firearm)	Recall (Firearm)	F1-Score (Firearm)	Precision (Non-Firearm)	Recall (Non-Firearm)	F1-Score (Non-Firearm)	Accuracy
Random Forest	0.85	0.86	0.86	0.74	0.72	0.73	0.81
XGBoost	0.73	0.92	0.82	0.71	0.36	0.48	0.72
CatBoost	0.73	0.92	0.81	0.69	0.36	0.47	0.72

4.3.2. Precision difference with and without SMOTE

	Precision (Firearm)	Recall (Firearm)	F1-Score (Firearm)	Precision (Non-Firearm)	Recall (Non-Firearm)	F1-Score (Non-Firearm)
NO SMOTE	0.73	0.73	0.73	0.45	0.46	0.45
WITH SMOTE	0.69	0.75	0.72	0.73	0.66	0.69