### **Model Documentation**

#### **Overview**

This project focuses on classifying security incidents into four categories: *Benign Positive*, *True Positive*, *False Positive*, and *Unknown*. The dataset consists of 4,758,418 entries and 45 features, including categorical and numerical fields. The target variable, IncidentGrade, is highly imbalanced, prompting the use of Synthetic Minority Oversampling Technique (SMOTE) to balance the classes.

#### **Rationale for Chosen Methods**

1. **Data Cleaning and Preprocessing**:
   * Columns with more than 50% missing values were dropped, leaving 31 features.
   * No columns with missing values between 5% and 50% were retained.
   * Features were scaled and encoded where applicable to ensure compatibility with machine learning models.
2. **Handling Imbalanced Classes**:
   * SMOTE was employed to upsample minority classes, ensuring equal representation of each category in the training data.
3. **Model Selection**:
   * **Decision Tree**: Chosen for interpretability and baseline performance evaluation.
   * **Logistic Regression**: Included to evaluate linear separability of the data.
   * **Random Forest**: Selected for its ensemble-based approach, high accuracy, and feature importance insights.
4. **Evaluation Metrics**:
   * Metrics used include accuracy, precision, recall, F1-score, and Macro F1-score to comprehensively assess model performance.
   * The Macro F1-score emphasizes balanced performance across all classes.

#### **Key Challenges and Solutions**

1. **High Dimensionality**:
   * Many columns had irrelevant or redundant data. Feature selection based on importance in Random Forest reduced overfitting and improved interpretability.
2. **Class Imbalance**:
   * The target distribution was heavily skewed towards *Benign Positive* and *True Positive*. SMOTE effectively balanced the dataset for training.
3. **Model Optimization**:
   * Grid search for hyperparameters in Random Forest and Decision Tree models enhanced accuracy and stability.
4. **Missing Data**:
   * Imputed or dropped columns with excessive missing data to prevent bias.

#### **Model Performance**

1. **Decision Tree**:
   * **Accuracy**: 97%
   * **Macro F1-Score**: 0.957
   * **Insights**: Strong performance but slightly prone to overfitting.
2. **Logistic Regression**:
   * **Accuracy**: 55%
   * **Macro F1-Score**: 0.446
   * **Insights**: Poor handling of non-linearity and class imbalance.
3. **Random Forest**:
   * **Accuracy**: 96%
   * **Macro F1-Score**: 0.957
   * **Insights**: The best-performing model, providing excellent precision, recall, and robustness.

**Feature Importance (Top 5 Features)**:

* OrgId: 19.5%
* IncidentId: 12.6%
* AlertId: 11.1%
* DetectorId: 10.9%
* AlertTitle: 9.2%

#### **Key Findings**

1. The dataset exhibits high correlation between IncidentId and the target variable, emphasizing its predictive value.
2. Random Forest outperformed other models, with robust results across all metrics.
3. SMOTE effectively addressed class imbalance, yielding a more balanced classification.

### **Recommendations**

1. **Integration into SOC Workflows**:
   * Deploy the Random Forest model as an automated alert classification tool.
   * Integrate the model into SIEM platforms for real-time incident prioritization.
   * Use feature importance to guide SOC analysts in investigating high-impact variables like OrgId and IncidentId.
2. **Future Improvements**:
   * Explore advanced techniques like XGBoost or LightGBM for further performance gains.
   * Incorporate temporal features such as timestamps for sequential pattern analysis.
   * Leverage additional external threat intelligence feeds to enrich model inputs.
3. **Deployment Considerations**:
   * Regularly retrain the model with updated datasets to account for evolving threat patterns.
   * Implement explainability tools (e.g., SHAP, LIME) for transparent predictions.
   * Monitor model performance in production, ensuring continuous alignment with SOC needs.

By automating incident classification and prioritization, the SOC can allocate resources efficiently, reducing response times and mitigating security risks.