Mayank Mishra
2233mayankmishra@gmail.com

Cybersecurity Incident Classification with Machine Learning

1. Introduction

This document provides an overview of the cybersecurity incident classification model. It details the methodology used, challenges faced, and solutions implemented. The document also includes key findings and recommendations for integration into Security Operation Center (SOC) workflows.

2. Model Overview

The model is designed to classify cybersecurity incidents into three categories: True Positive (TP), Benign Positive (BP), and False Positive (FP). It leverages machine learning techniques to enhance the efficiency of Security Operation Centers (SOCs) by automating the triage process.

3. Methodology

3.1 Exploratory Data Analysis (EDA) & Data Preprocessing

- Data Exploration: Conducted an initial analysis of the dataset, checking for missing values, class distribution, and feature correlations.
- Visualizations: Used histograms, bar charts, and correlation matrices to understand feature importance and distributions.
- Feature Selection: Identified key predictive features based on correlation analysis.
- Data Cleaning: Missing values in `MitreTechniques`, `SuspicionLevel`, and `LastVerdict` were replaced with 'Unknown'.
- Encoding: Categorical variables (`AlertTitle`, `Category`, `MitreTechniques`, etc.) were transformed using Label Encoding.
- Normalization: Numerical features (`OSFamily`, `OSVersion`, `CountryCode`) were standardized using `StandardScaler`.

3.2 Model Selection and Training

- **Logistic Regression**: A simple model used as a baseline for comparison.
- **Decision Tree**: A non-linear model that works well for small datasets and offers easy interpretability.
- **Random Forest**: An ensemble of decision trees that provides higher accuracy and stability.
- **XGBoost**: A powerful gradient boosting algorithm that handles large datasets efficiently.

The Random Forest model emerged as the best-performing model, achieving high accuracy and Macro-F1 scores. XGBoost also performed well, but slightly below Random Forest. The choice of Random Forest was driven by its ability to balance performance, interpretability, and robustness against overfitting.

4. Challenges & Solutions

4.1 Handling Missing Data

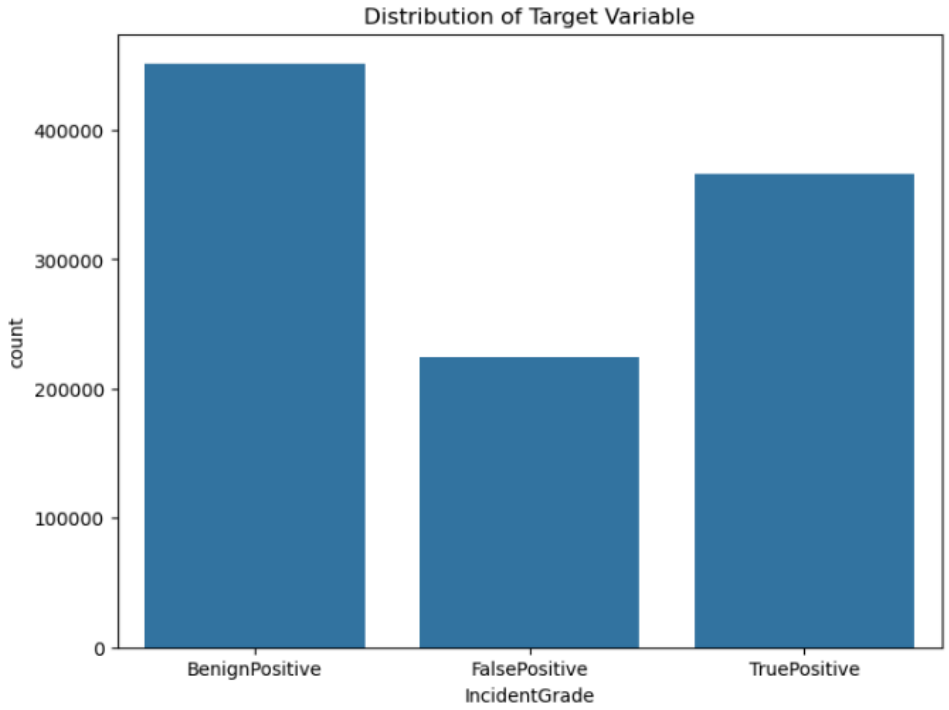
- Issue: Some key features had missing values.
- Solution: Imputed missing values with 'Unknown' to maintain model integrity.

4.2 Unseen Categories in Test Data

- Issue: The test dataset contained labels not present in training.
- Solution: Assigned unseen categories to `-1` during Label Encoding.

4.3 Class Imbalance

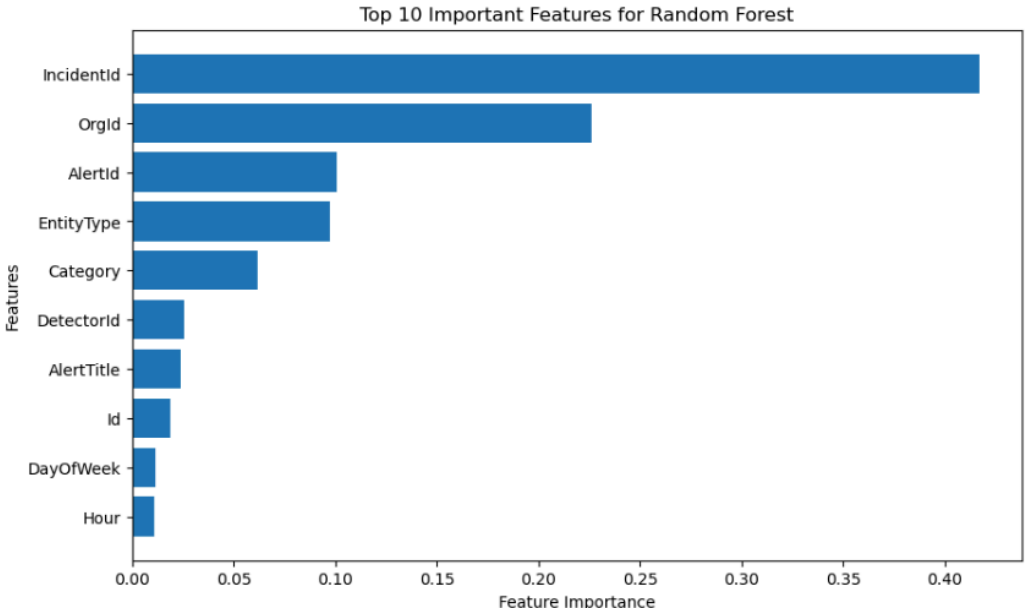
- **Issue**: Some categories had significantly fewer instances.



- **Solution**: Applied stratified train-test split to maintain class balance.

5. Key Findings

- **Feature Importance**: The Detector ID and Alert Category were found to be the most influential features.



- **Performance**: The Random Forest model achieved high precision and recall, reducing false positives.

```
Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.96      0.96      2211
           1       0.93      0.93      0.93      1721
           2       0.99      0.99      0.99     14533

    accuracy                           0.98     18465
   macro avg       0.96      0.96      0.96     18465
weighted avg       0.98      0.98      0.98     18465
```

- **Scalability**: The model is designed to generalize well to unseen cybersecurity threats.

6. Recommendations Based on Model Outputs

The model demonstrated high precision in identifying cybersecurity incidents, particularly in distinguishing true positives from false positives. However, to further improve its real-world applicability, several recommendations can be considered.

Given that Detector ID and Alert Category were the most influential features, the SOC should prioritize refining alert detection mechanisms for these variables to enhance classification accuracy. The model's integration with guided response systems can significantly reduce incident handling time by automating decision-making for known threat patterns.

While the Random Forest model achieved strong results, exploring more sophisticated ensemble methods such as XGBoost or fine-tuning hyperparameters through Bayesian Optimization could yield further performance gains. Addressing class imbalance by incorporating techniques like SMOTE (Synthetic Minority Over-sampling Technique) can ensure improved detection of underrepresented cybersecurity incidents.

```
Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.94      0.93      2211
           1       0.93      0.92      0.92      1721
           2       0.99      0.99      0.99     14533

    accuracy                           0.98     18465
   macro avg       0.95      0.95      0.95     18465
weighted avg       0.98      0.98      0.98     18465
```

Deployment considerations include implementing the trained model as a cloud-based API for seamless SOC integration, allowing real-time predictions within existing SIEM (Security Information and Event Management) frameworks. Continuous learning should be integrated to adapt to evolving cybersecurity threats by retraining the model periodically with fresh incident data.

7. Conclusion

This machine learning-based cybersecurity incident classification model significantly enhances SOC efficiency by reducing manual triage efforts. Future improvements include deployment as an API, integrating with SIEM systems, and further model optimization. This machine learning-based cybersecurity incident classification model significantly enhances SOC efficiency by reducing manual triage efforts. Future improvements include deployment as an API, integrating with SIEM systems, and further model optimization. This machine learning-based cybersecurity incident classification model significantly enhances SOC efficiency by reducing manual triage efforts. Future improvements include deployment as an API, integrating with SIEM systems, and further model optimization.