Project Documentation Report

Project Title

Microsoft: Classifying Cybersecurity Incidents with Machine Learning

Domain

Cybersecurity and Machine Learning

Problem Statement

As a data scientist at Microsoft, the objective is to enhance the efficiency of Security Operation Centers (SOCs) by developing a machine learning model that accurately predicts the triage grade of cybersecurity incidents. Utilizing the GUIDE dataset, the model will categorize incidents as true positive (TP), benign positive (BP), or false positive (FP) based on historical evidence and customer responses. This model will support guided response systems, providing SOC analysts with precise, context-rich recommendations to improve the security posture of enterprise environments.

The model will be trained on the train.csv dataset and evaluated using macro-F1 score, precision, and recall metrics on the test.csv dataset to ensure reliable performance in real-world applications.

Skills Takeaways

- **Data Preprocessing and Feature Engineering**
- **Machine Learning Classification Techniques**
- **Model Evaluation Metrics (Macro-F1 Score, Precision, Recall)**
- **Cybersecurity Concepts and Frameworks (MITRE ATT&CK)**
- **Handling Imbalanced Datasets**
- **Model Benchmarking and Optimization**

Business Use Cases

The developed solution can be implemented in various scenarios within the field of cybersecurity, including:

- **Security Operation Centre (SOCs):** Automating the triage process by accurately classifying cybersecurity incidents, enabling SOC analysts to prioritize critical threats more efficiently.
- **Incident Response Automation:** Enhancing guided response systems to suggest appropriate actions for different types of incidents, leading to quicker mitigation of threats.
- **Threat Intelligence:** Improving threat detection capabilities by integrating historical evidence and customer responses into the triage process.
- **Enterprise Security Management:** Strengthening the security posture of enterprise environments by minimizing false positives and ensuring prompt responses to true threats.

Approach

Data Exploration and Understanding

1. **Initial Inspection:**

- Load the train.csv dataset.
- Inspect the structure of the data, including feature types and distributions of the target variable (TP, BP, FP).

2. **Exploratory Data Analysis (EDA):**
    - Perform visualizations and statistical summaries to identify patterns, correlations, and anomalies.
    - Address class imbalances that may require specific handling strategies.

Data Preprocessing

1. **Handling Missing Data:**
    - Identify missing values and decide on imputation, removal, or handling strategies based on model requirements.
2. **Feature Engineering:**
    - Create or modify features to enhance model performance.
    - Example: Extract temporal features from timestamps (hour of day, day of week) or normalize numerical features.
3. **Encoding Categorical Variables:**
    - Convert categorical features to numerical using one-hot encoding, label encoding, or target encoding based on the relationship with the target variable.

Data Splitting

1. **Train-Validation Split:**
    - Split train.csv into training and validation sets using a standard ratio (e.g., 70-30 or 80-20).
2. **Stratification:**
    - Ensure that training and validation sets have similar class distributions through stratified sampling if the target variable is imbalanced.

Model Selection and Training

1. **Baseline Model:**
    - Begin with a simple model (e.g., logistic regression or decision tree) to establish a performance benchmark.
2. **Advanced Models:**
    - Experiment with models like Random Forests, Gradient Boosting Machines (XGBoost, LightGBM), and Neural Networks.
    - Tune models using grid search or random search over hyperparameters.
3. **Cross-Validation:**

    - Implement k-fold cross-validation to ensure consistent model performance across different data subsets.

Model Evaluation and Tuning

1. **Performance Metrics:**
    - Evaluate the model using the validation set, focusing on macro-F1 score, precision, and recall.
    - Ensure balanced performance across TP, BP, and FP classes.
2. **Hyperparameter Tuning:**
    - Fine-tune hyperparameters to optimize model performance.
3. **Handling Class Imbalance:**
    - Use techniques like SMOTE, adjusting class weights, or ensemble methods to handle class imbalances effectively.

Model Interpretation

1. **Feature Importance:**
    - Analyse feature importance using SHAP values, permutation importance, or model-specific methods.
2. **Error Analysis:**
    - Identify common misclassifications to refine the model or improve feature engineering.

Final Evaluation on Test Set

1. **Testing:**
    - Evaluate the optimized model on the test.csv dataset, reporting the final macro-F1 score, precision, and recall.
2. **Comparison to Baseline:**
    - Compare test set performance to the baseline model and initial validation results.

Documentation and Reporting

1. **Model Documentation:**
    - Document the entire process, including methods chosen, challenges faced, and how they were addressed.
2. **Recommendations:**
    - Suggest how the model can be integrated into SOC workflows, potential improvements, and deployment considerations.

Results

- **Machine Learning Model:** A model capable of predicting the triage grade of cybersecurity incidents with high macro-F1 score, precision, and recall.
- **Performance Analysis:** Comprehensive insights into model performance, including influential features.
- **Documentation:** Detailed report covering data preprocessing, model selection, evaluation, and deployment strategies.

Project Evaluation Metrics

- **Macro-F1 Score:** Measures balanced performance across TP, BP, and FP classes.
- **Precision:** Accuracy of positive predictions, crucial for minimizing false positives.
- **Recall:** Model's ability to correctly identify true positives, essential for ensuring no real threats are missed.

Technical Tags

- Machine Learning
- Classification
- Cybersecurity
- Data Science
- Model Evaluation
- Feature Engineering
- SOC
- Threat Detection

Dataset Overview

The GUIDE dataset is organized into three hierarchical levels: evidence, alert, and incident. The dataset's primary objective is to predict incident triage grades based on historical customer responses. It includes a training dataset with 45 features across 1 million triage-annotated incidents. The dataset is divided into train and test sets (70% and 30%, respectively), ensuring relevance across different levels.

Data Set Explanation

The dataset consists of records of cybersecurity incidents, with preprocessing steps including handling missing data, feature engineering, and normalization/standardization.

Project Deliverables

- **Source Code:** Well-documented code from data preprocessing to model evaluation.
- **Model File:** The trained machine learning model ready for deployment.
- **Documentation:** A comprehensive report covering the entire project.

- **Presentation:** A summary presentation highlighting project outcomes and potential business impacts.

This documentation provides a clear and detailed overview of the project, making it easier for stakeholders to understand the approach, methodologies, and results achieved.

4o