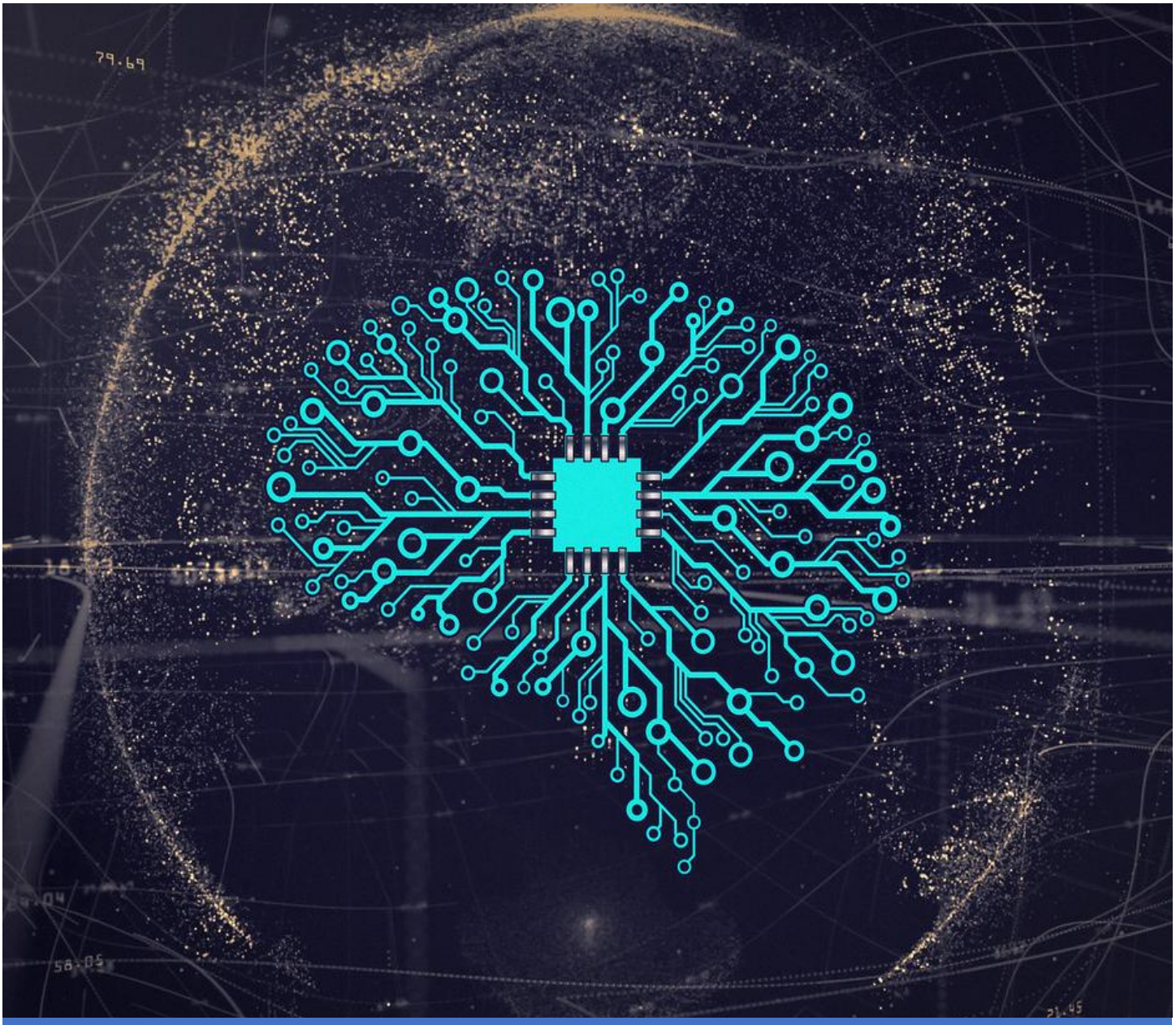


Microsoft: Classifying Cybersecurity Incidents with Machine Learning

BY : Aashifa

1. Project Overview

1.1 Problem Statement

The goal of this project is to develop a machine learning model to predict the triage grade of cybersecurity incidents for Security Operation Centers (SOCs). These incidents are categorized into:

- **True Positive (TP)**
- **Benign Positive (BP)**
- **False Positive (FP)**

The classification is based on historical evidence and customer responses, enabling SOC analysts to focus on accurate recommendations and improve enterprise security posture.

1.2 Objectives

- **Incident Classification:** Predict whether an incident is TP, BP, or FP.
- **Model Development:** Develop a robust, scalable, and interpretable model for incident classification.
- **Evaluation Metrics:** Use precision, recall, and F1-score to evaluate model performance.

1.3 Expected Outcomes

- A high-performing machine learning model for triage classification.
- Insights into influential features for incident classification.
- An evaluation of model strengths and weaknesses.

2. Data Collection and Preprocessing

2.1 Data Source

The dataset used in this project is the GUIDE train dataset and GUIDE test dataset, which contains historical cybersecurity incident data. This data includes information about various incident characteristics, such as:

- **Incident Type**: Describes the type of incident (e.g., InitialAccess, Exfiltration, CommandAndControl).
- **Incident Severity**: Categorized into IncidentGrade (e.g., TruePositive, FalsePositive, BenignPositive).
- **Response Time**: Timestamp for when the alert was generated, indicating the response time.

- **Affected Systems**: Various system attributes, such as OSFamily, OSVersion, ResourceIdName, etc.
- **Customer Feedback**: The classification of the incident based on customer responses, such as TruePositive, FalsePositive, or BenignPositive.

2.2 Preprocessing Overview

Initial Dataset Shape:

- **Train Data**: (9,516,837 rows, 45 columns)
- **Test Data**: (4,147,992 rows, 46 columns)

Key Preprocessing Steps:

Step	Action Taken	Description
1	Handling duplicates	Removed duplicates from the training and test data.
2	Handling Missing Values	Imputed or dropped missing data depending on context.
3	Encoding Categorical Variables	One-hot encoding applied to features like **Incident Grade**
4	Normalization	Scaled continuous variables such as **Time Stamp**
5	Feature selection	Retained relevant features after correlation analysis

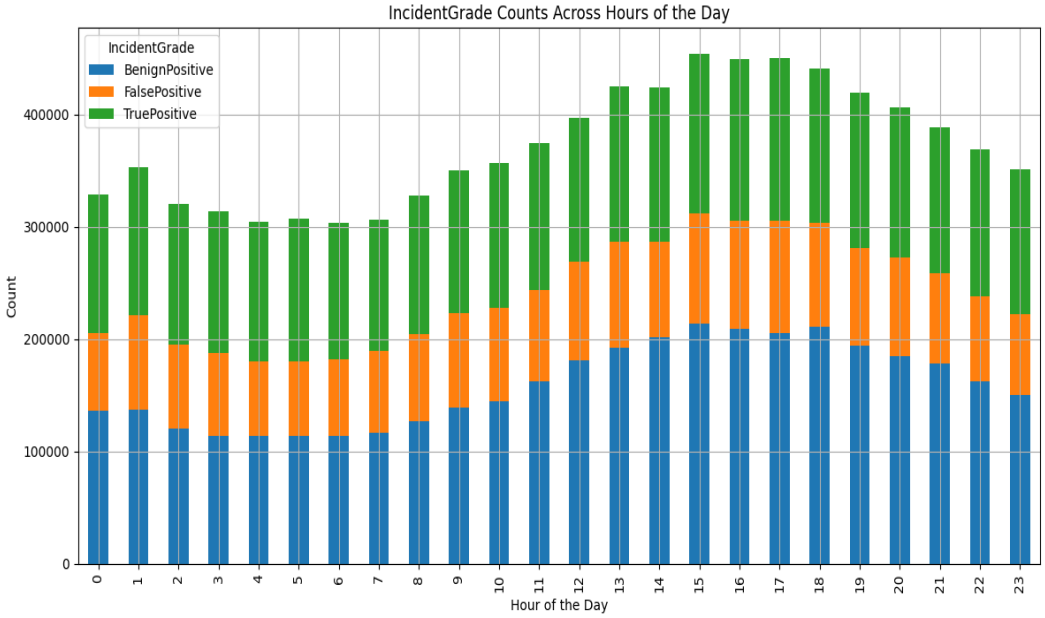
Results after Preprocessing

Dataset	Duplicate Rows Removed	Final Shape
Train	542,692	(8,922,805, 35)
Test	104	(4,147,888, 46)

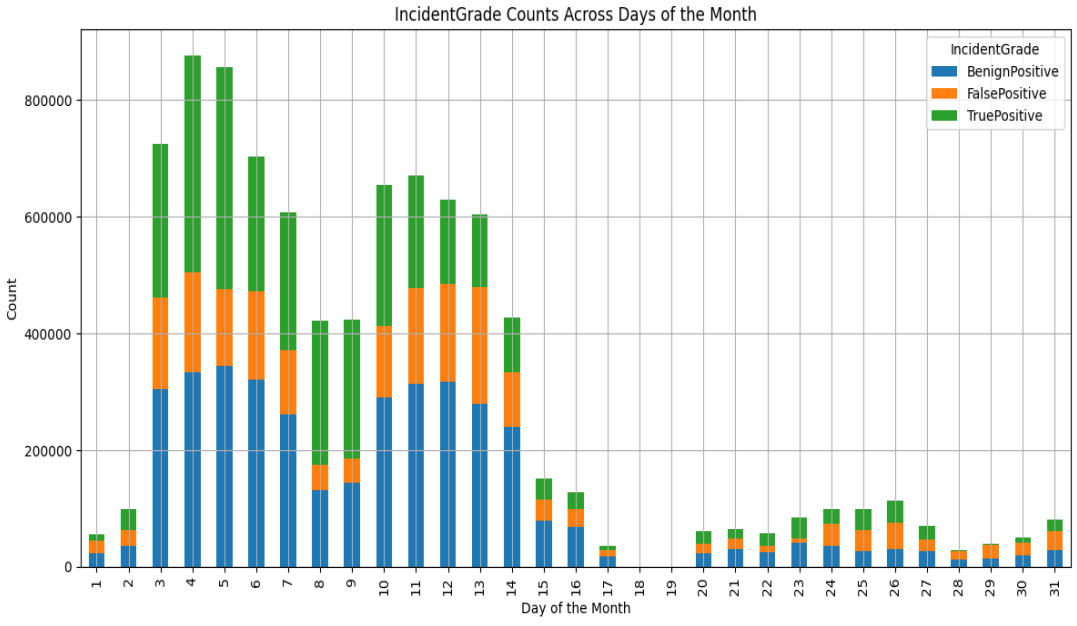
2.3 Exploratory Data Analysis

To gain insights into the data, the following exploratory analyses were conducted:

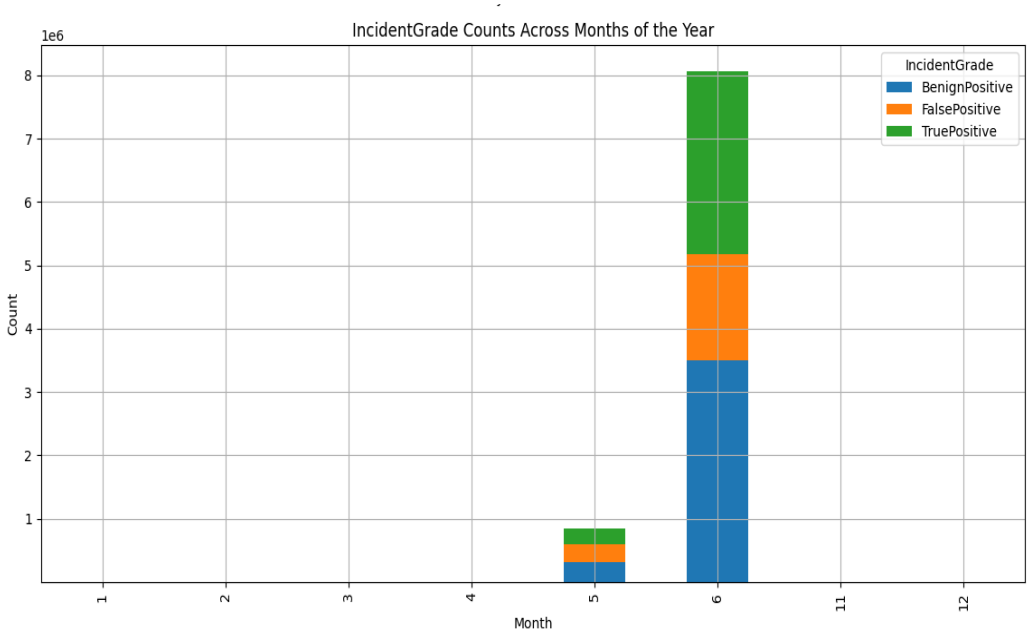
1. **Distribution of Incident Grades Across Time**:
    - Three visualizations were created to observe how IncidentGrade is distributed across:
        - **Hours of the Day**: Displays hourly patterns, highlighting peak hours for incidents.



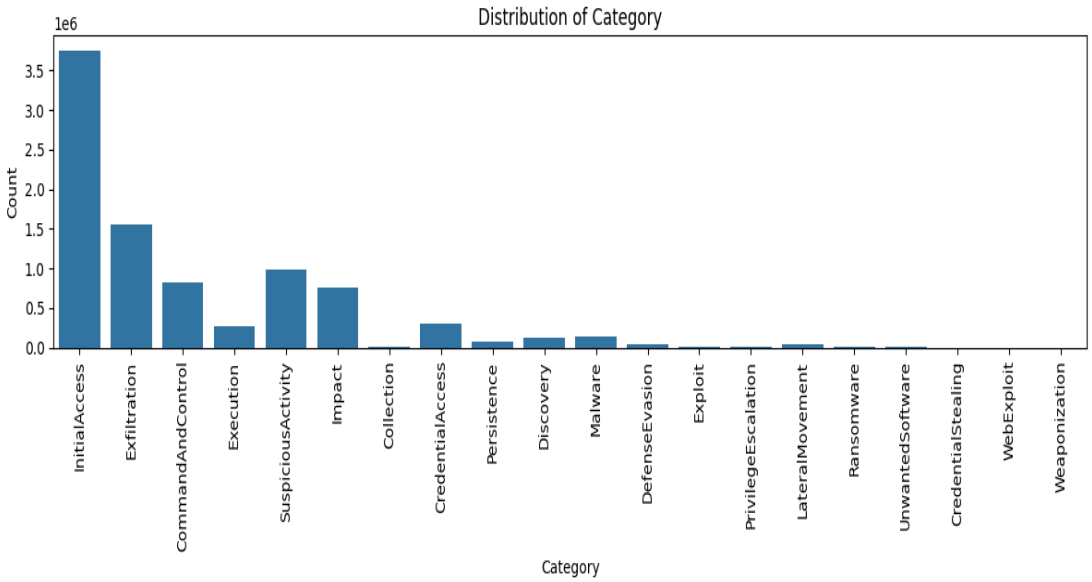
- **Days of the Month**: Shows how incidents are distributed on specific days.



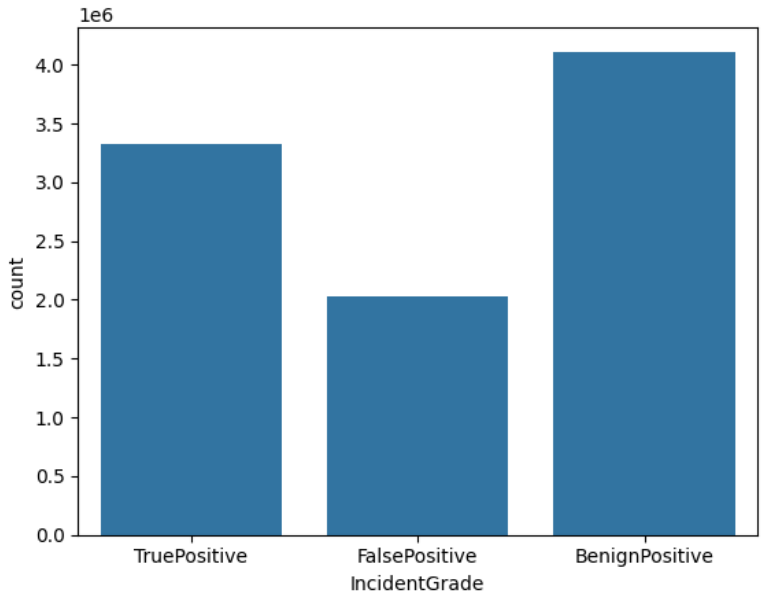
- **Months of the Year**: Illustrates monthly trends, emphasizing incident counts.



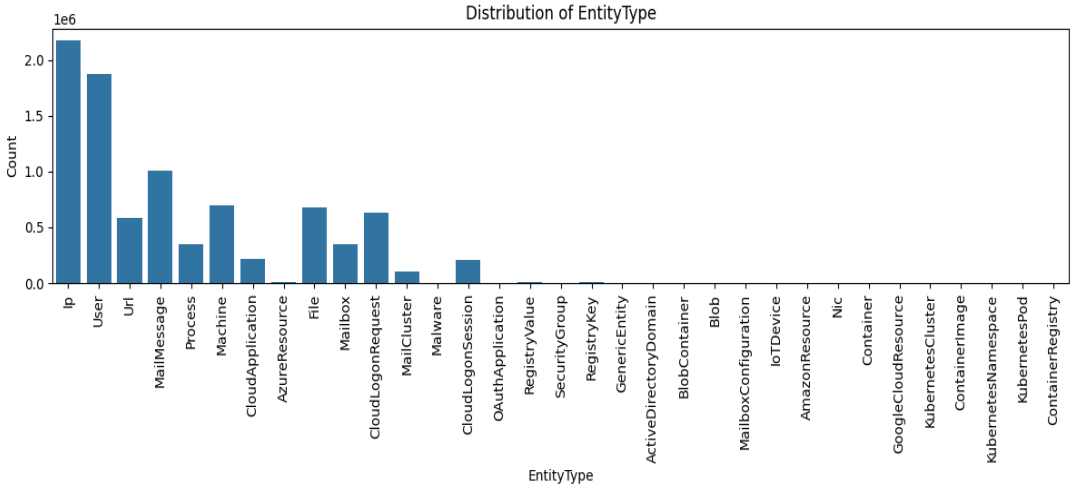
2. **Distribution of Key Attributes**:
    - Bar plots were used to analyze:
        - Category: The most common incident categories include InitialAccess and Exfiltration.



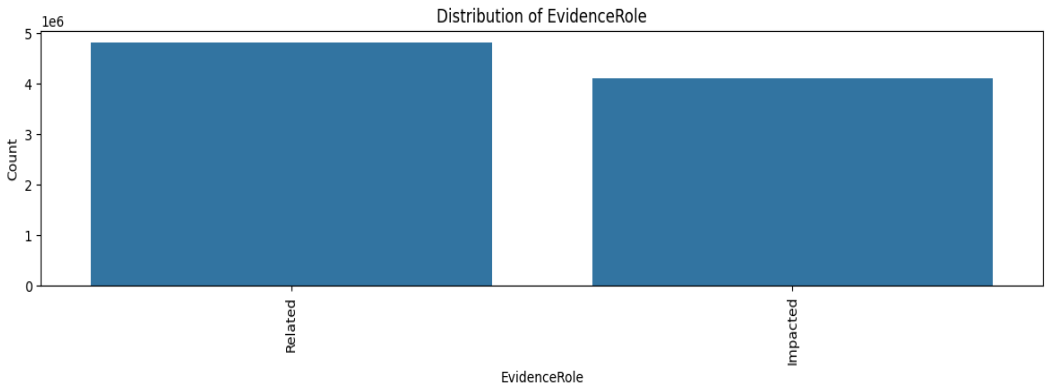
        - IncidentGrade: BenignPositive incidents are the most frequent, followed by TruePositive.



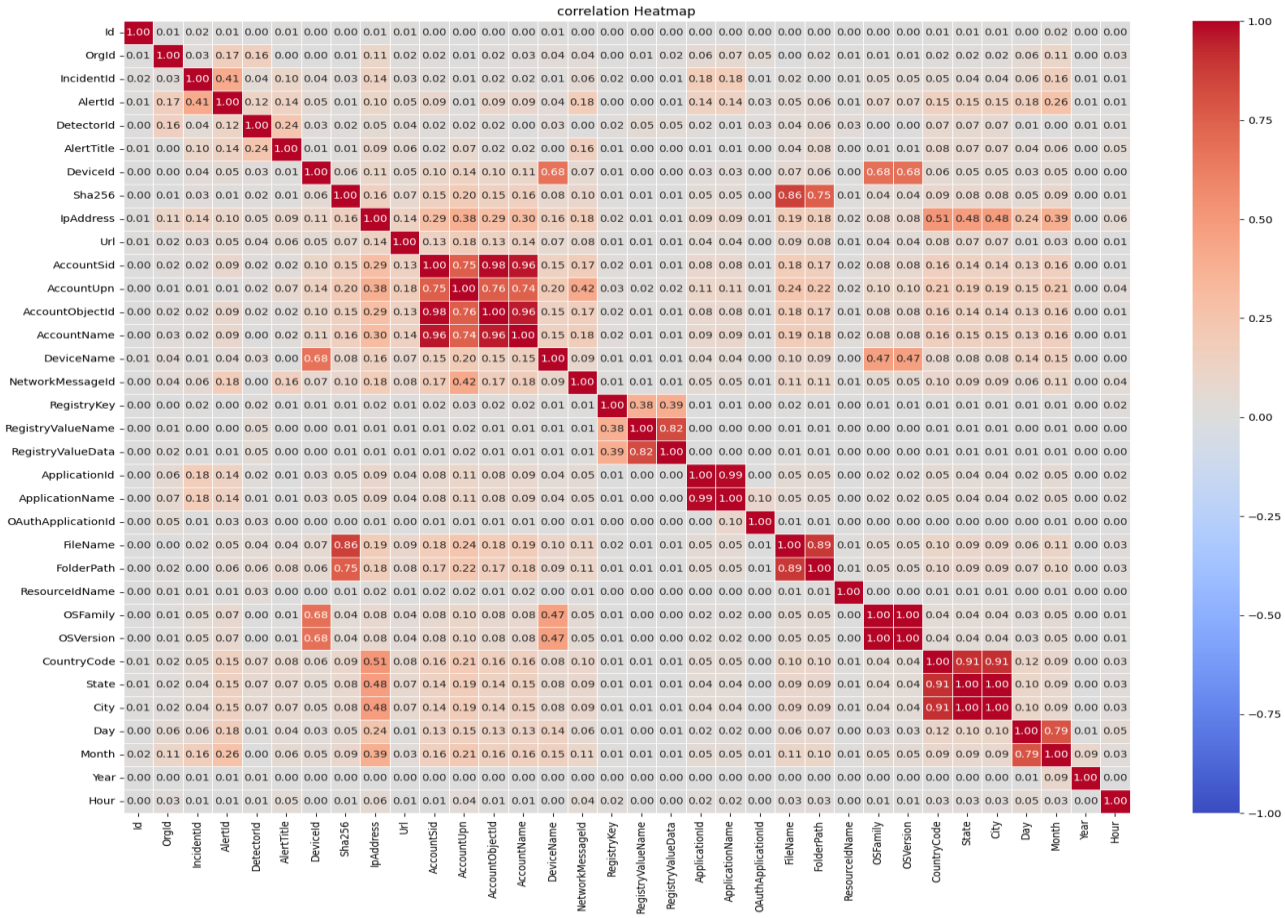
- EntityType: Significant entities include IP, User, and URL.



- EvidenceRole: The dataset has a balanced distribution between Related and Impacted evidence roles.



3. **Correlation Analysis**:
    - A heatmap was generated to explore correlations between numerical variables. Key observations include:
        - High correlation between RegistryKey and RegistryValueData.
        - Temporal variables (Day, Month, Hour) show patterns indicating time-based trends.



3. Model Selection and Comparison

3.1 Model Selection Criteria

The models were selected based on the following criteria:

- **Simplicity and Interpretability**: Models that can be easily understood and explained to non-technical stakeholders.
- **Performance**: Models capable of accurately predicting the triage grade of incidents.
- **Scalability**: Models efficient enough to handle large datasets in a production environment.
- **Generalization**: Models that avoid overfitting and perform consistently well on unseen data.

3.2 Models Tested

1. **Logistic Regression**
    - **Reason for Selection**: Serves as a baseline model to assess the effectiveness of a linear approach.
    - **Assumptions**: Assumes a linear relationship between input features and the target variable.
2. **Decision Tree**
    - **Reason for Selection**: Captures non-linear relationships in the data and provides easy-to-interpret feature importance.
    - **Assumptions**: Assumes data can be segmented into distinct decision regions corresponding to classes.
3. **Random Forest**
    - **Reason for Selection**: An ensemble method combining multiple decision trees to reduce overfitting and improve accuracy.
    - **Assumptions**: Averaging multiple trees enhances prediction accuracy and generalization.
4. **LightGBM**
    - **Reason for Selection**: Known for high performance and speed on large datasets, using histogram-based learning for efficiency.
    - **Assumptions**: Boosting weak models iteratively results in a robust final model.
5. **XGBoost**
    - **Reason for Selection**: A powerful gradient-boosting method with strong performance on imbalanced datasets and built-in regularization to prevent overfitting.
    - **Assumptions**: Iterative boosting improves accuracy, and regularization ensures robustness.

3.3 Model Evaluation Metrics

The models were evaluated using the following metrics:

- **Accuracy**: Proportion of correctly classified incidents.
- **Precision**: Measure of the quality of positive predictions for each class (TP, BP, FP).
- **Recall**: Proportion of actual incidents correctly identified for each class.
- **F1-Score**: Harmonic mean of precision and recall, balancing the two.

3.4 Performance Comparison

Model	Accuracy	Precision (TP)	Precision (BP)	Precision (FP)	Recall (TP)	Recall (BP)	Recall (FP)	F1-Score (TP)	F1-Score (BP)	F1-Score (FP)
Logistic Regression	50%	0.54	0.33	0.53	0.53	0.26	0.62	0.54	0.29	0.57
Decision Tree	100%	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00
Random Forest	82%	0.76	0.79	0.97	0.93	0.73	0.75	0.84	0.76	0.85
LightGBM	87%	0.88	0.75	0.96	0.88	0.89	0.83	0.88	0.81	0.89
XGBoost	90%	0.84	0.96	0.97	0.98	0.79	0.88	0.91	0.87	0.92

3.5 Insights from Comparison

- **Logistic Regression:** Provides a baseline model with reasonable accuracy but struggles with imbalanced data resulting in lower F1-scores for TP and BP. While interpretable and simple, it falls short in capturing data complexity compared to ensemble models.
- **Decision Tree**: Perfectly fits the training set but shows signs of overfitting, struggling to generalize on the test set.
- **Random Forest**: Achieves balanced performance but falls short of XGBoost in precision and recall for TP and BP.
- **LightGBM**: Combines speed and accuracy effectively, achieving competitive performance (87% accuracy).
- **XGBoost**: Outperforms other models, delivering 90% accuracy and the highest F1-scores across all classes. Its ability to handle imbalanced data and provide robust performance makes it the preferred model.

3.6 Feature Importance and Model Insights

XGBoost's feature importance analysis highlights the most influential factors in incident classification. This enables SOC analysts to focus on critical attributes, improving their decision-making process.

4. Model Evaluation and Results

The final evaluation of the best model, XGBoost, was done using the test set. The evaluation metrics for the model on the test set are as follows:

- **Accuracy**: 90%
- **Macro avg F1-score**: 90%
- **Weighted avg F1-score**: 90%

These results indicate that the model performs exceptionally well in classifying incidents accurately and providing reliable predictions for SOC analysts.

5. Conclusion

In this project, we successfully developed and evaluated a machine learning model to predict the triage grade of cybersecurity incidents. The XGBoost model outperformed other models in terms of accuracy and F1-score. This model provides actionable insights for SOC analysts and helps improve the overall security posture of enterprise environments.