**Classifying Cybersecurity Incidents with Machine Learning Report**

**1.Project Objective:**

The objective of this project is to classify cybersecurity incidents. My aim is to developing the machine learning model that categorizes incidents as true positive (TP), benign positive (BP), or false positive (FP) that accurately predict incidents.

**2.Data Preprocessing:**

**Data Cleaning:**

Describe steps taken to clean the data, including handling missing values, removing duplicates, or correcting data inconsistencies.

**Feature Engineering:**

Create new features or modify existing ones to improve model performance. For example, combining related features, deriving new features from timestamps (like hour of the day or day of the week), or normalizing numerical variables.

**Encoding Categorical Variables:**

Convert categorical features into numerical representations using techniques like one-hot encoding, label encoding, or target encoding, depending on the nature of the feature and its relationship with the target variable.

**Train-Test Split:** Mention the split ratio and the criteria for dividing the data (e.g., 80-20 split for train-test data).

**Sampling Strategy:**

Document any resampling techniques used to address class imbalance (e.g., Random Oversampling, SMOTE, or under sampling).

**Scaling/Normalization:**

Describe the scaling process (StandardScaler) applied to ensure that feature values are standardized for model training.

**3.Exploratory Data Analysis (EDA)**

* **Feature Distribution:** Summarize the distribution of key features, showing histograms or box plots if relevant.
* **Correlation Analysis:** Present the correlation matrix or heatmap to highlight relationships between features.
* **Insights:** Include any significant insights found during EDA that influenced model selection.

**4.Model Selection and Training:**

Several models were trained to compare performance:

* **Baseline Models**: Logistic Regression, K-Nearest Neighbours (KNN), Decision Tree.
* **Advanced Models**: XGBoost, LightGBM, and Random Forest.

Each model was evaluated on metrics such as accuracy, precision, recall, and F1 score.

**Best Model:** Gradient Boosting Model (e.g., XGBoost/LightGBM, based on the test performance).

**Hyperparameter Tuning:** Performed hyperparameter tuning with GridSearchCV/RandomizedSearchCV to optimize model parameters such as:

* estimators: [e.g., 50, 100]
* learning rate: [e.g., 0.01, 0.05, 0.1]
* max\_depth: [e.g., 3, 5, 10]

**5.Model Evaluation**

* **Evaluation Metrics:** Accuracy, Precision, Recall, F1 Score were used to assess model performance.
* **Train vs Test Performance:**

| **Metric** | **Train Score** | **Test Score** |
| --- | --- | --- |
| Accuracy | [Train accuracy] | [Test accuracy] |
| Precision | [Train precision] | [Test precision] |
| Recall | [Train recall] | [Test recall] |
| F1 Score | [Train F1] | [Test F1] |

**6.Best Model:**

LightGBM provided the best results, balancing accuracy with computational efficiency.

**7.Feature Importance Analysis**

* **Top 10 Features:** Identified the 10 most important features contributing to the model's predictions.
* **Feature Importance Plot:** A bar chart showed the impact of each feature, indicating the top predictors.

**8. Conclusion**

The project successfully developed a classification model for cybersecurity incidents. The final model, optimized for accuracy and balanced class predictions, provides a reliable tool for incident categorization, potentially improving response prioritization and efficiency in cybersecurity operations.