CAPSTONE PROJECT4

By

S.Vinodha

Microsoft Classifying Cybersecurity Incidents

Machine Learning

Objective

to create a classification model that categorizes incidents based on historical evidence and customer responses as

true positive (TP),

benign positive (BP),

or false positive (FP)

Data Processing

Data Cleaning:

- Removed rows with NaN values.
- Removed duplicate rows.
- Synchronized columns between training and testing datasets.
- Replaced uncommon values with "Others" to standardize categorical features.

Balancing Classes:

- Under sampled majority classes in the dataset to ensure class balance using the minority class size.

Feature Transformation

- Ensured feature alignment between training and test datasets.

Model Training

Feature Selection:

Selected top features based on importance derived from a Random Forest model.

Models Trained:

Logistic Regression

Random Forest

Support Vector Classifier (SVM)

Encoding:

For categorical columns, (e.g., One-Hot Encoding) before training models.

Top Features:

Focused on a preselected list of top 10+ features for final model training.

Model	Accuracy	Precision Class 0	Recall Class 0	F1-Score Class 0	Precision Class 1	Recall Class 1	F1-Score Class 1	Precision Class 2	Recall Class 2	F1-Score Class 2
Logistic Regression	0.746184	0.667739	0.823413	0.737450	0.800459	0.707552	0.751143	0.802002	0.707209	0.751629
Random Forest	0.749171	0.664286	0.830357	0.738095	0.868455	0.672580	0.758069	0.765152	0.743011	0.753919
SVM	0.749171	0.664286	0.830357	0.738095	0.868455	0.672580	0.758069	0.765152	0.743011	0.753919

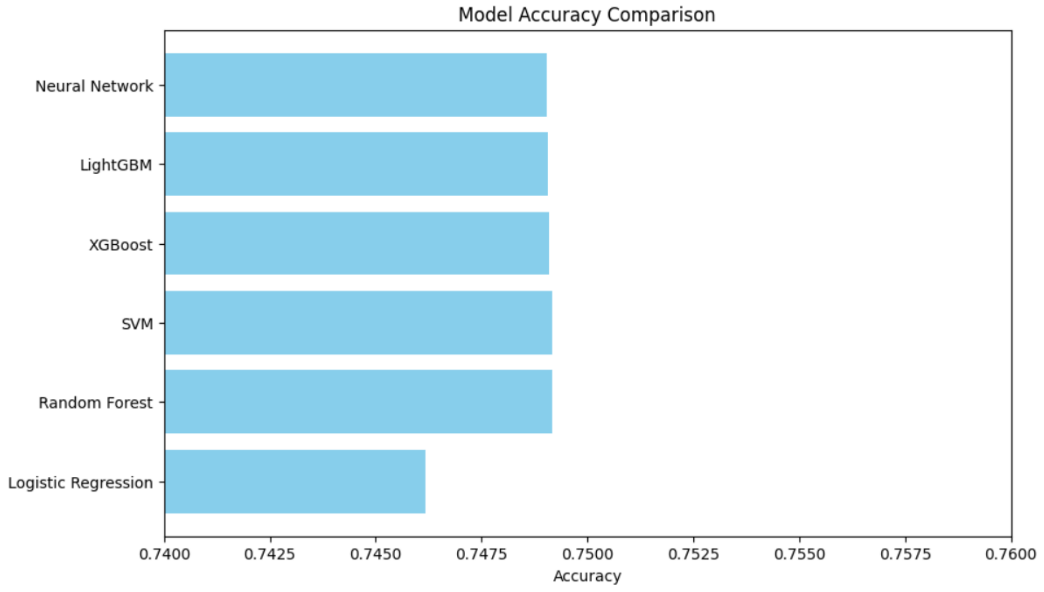
Evaluation

Model	Mean Accuracy (CV)	Standard Deviation (CV)
Logistic Regression	0.737640	0.002005
Random Forest	0.742252	0.002881
SVM	0.742153	0.002563

Cross Validation

Hyperparameter tuning

Model	Best Parameters
Logistic Regression	$C = 10$, Solver: lbfgs
Random Forest	No max depth, Min samples per leaf = 4, Min samples per split = 2
SVM	$C = 10$, Kernel: rbf, Gamma: scale



Model	Accuracy
Logistic Regression	0.746184
Random Forest	0.749171
SVM	0.749161
XGBoost	0.749101
LightGBM	0.749071
Neural Network	0.749036

Random Forest – best model

Feature Engineering

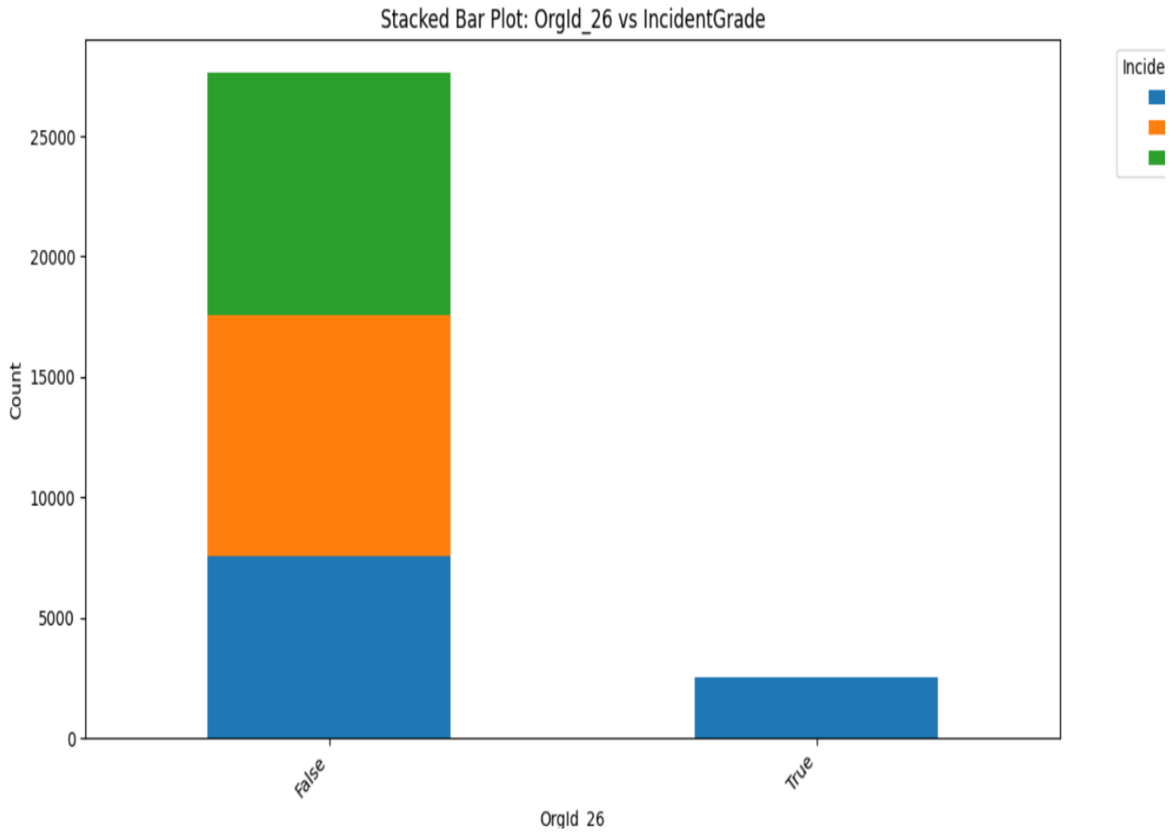
MitreTechniques - T1027;T1027.002;T1027.005;T1105;T1204.002

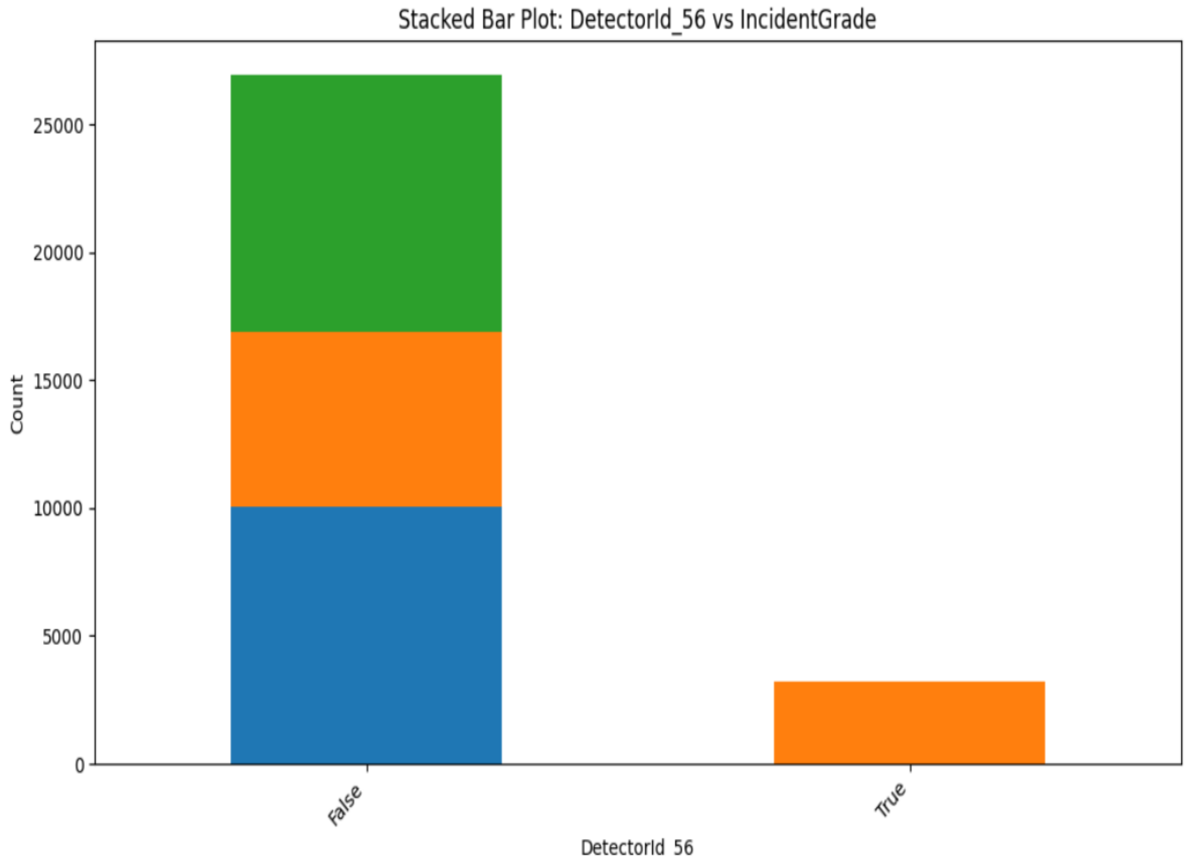
- MitreTechniques _T1027
- MitreTechniques _T1027.002
- MitreTechniques _T1027.005
- MitreTechniques _T1105
- MitreTechniques _T1204.002

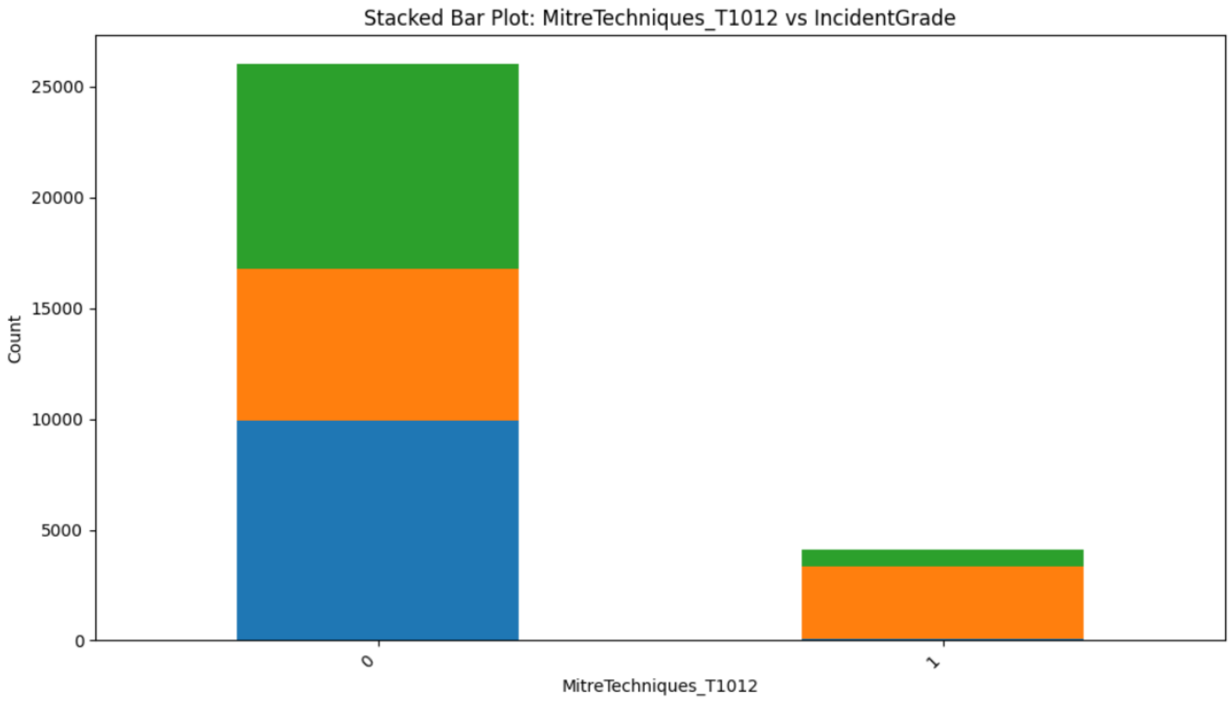
Smote – for unbalanced data

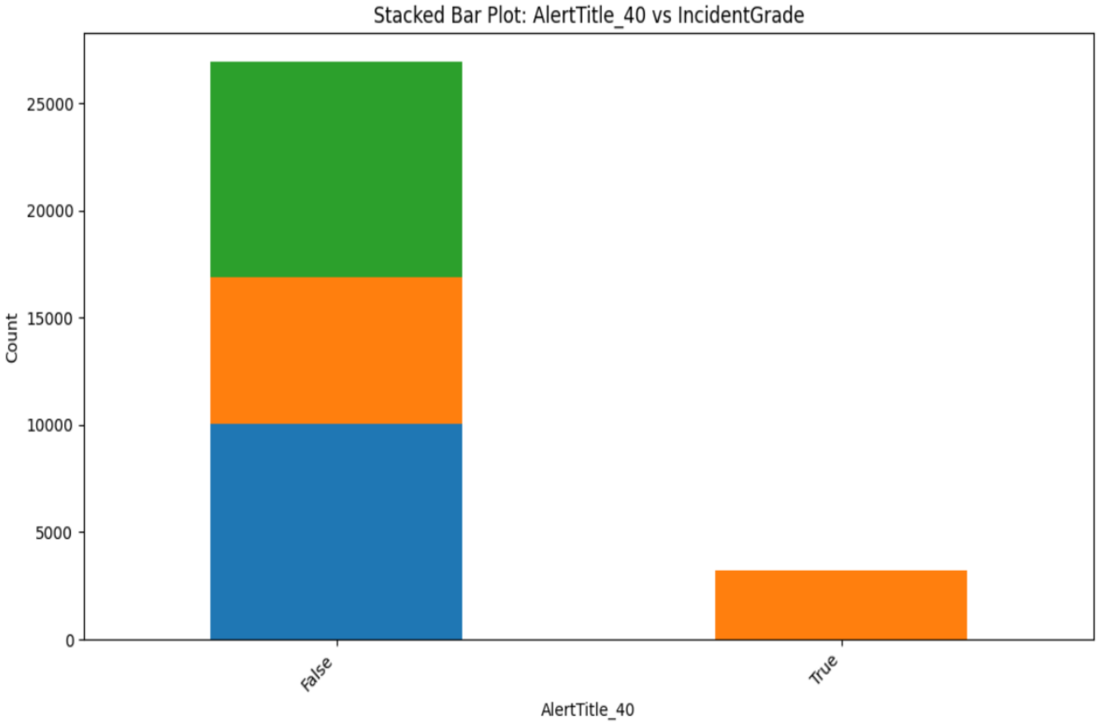
The results suggest that Random Forest are more robust after SMOTE

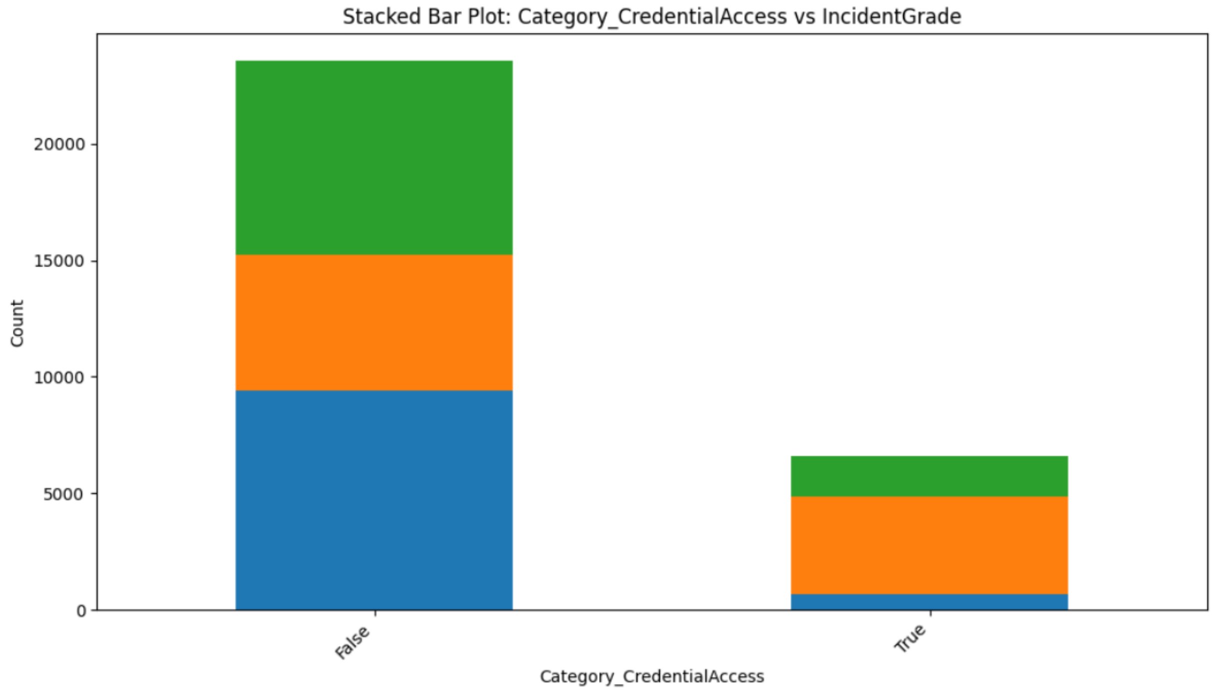
Visualizing selected features

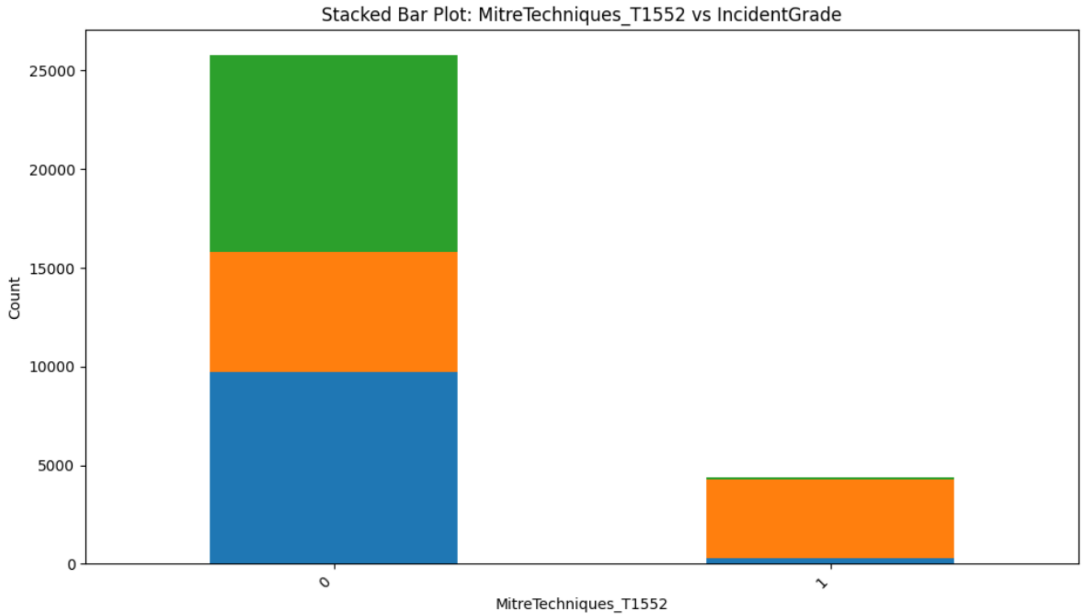





Stacked Bar Plot: MitreTechniques_T1012 vs IncidentGrade
Count
25000
20000
15000
10000
5000
0
0
1
MitreTechniques_T1012

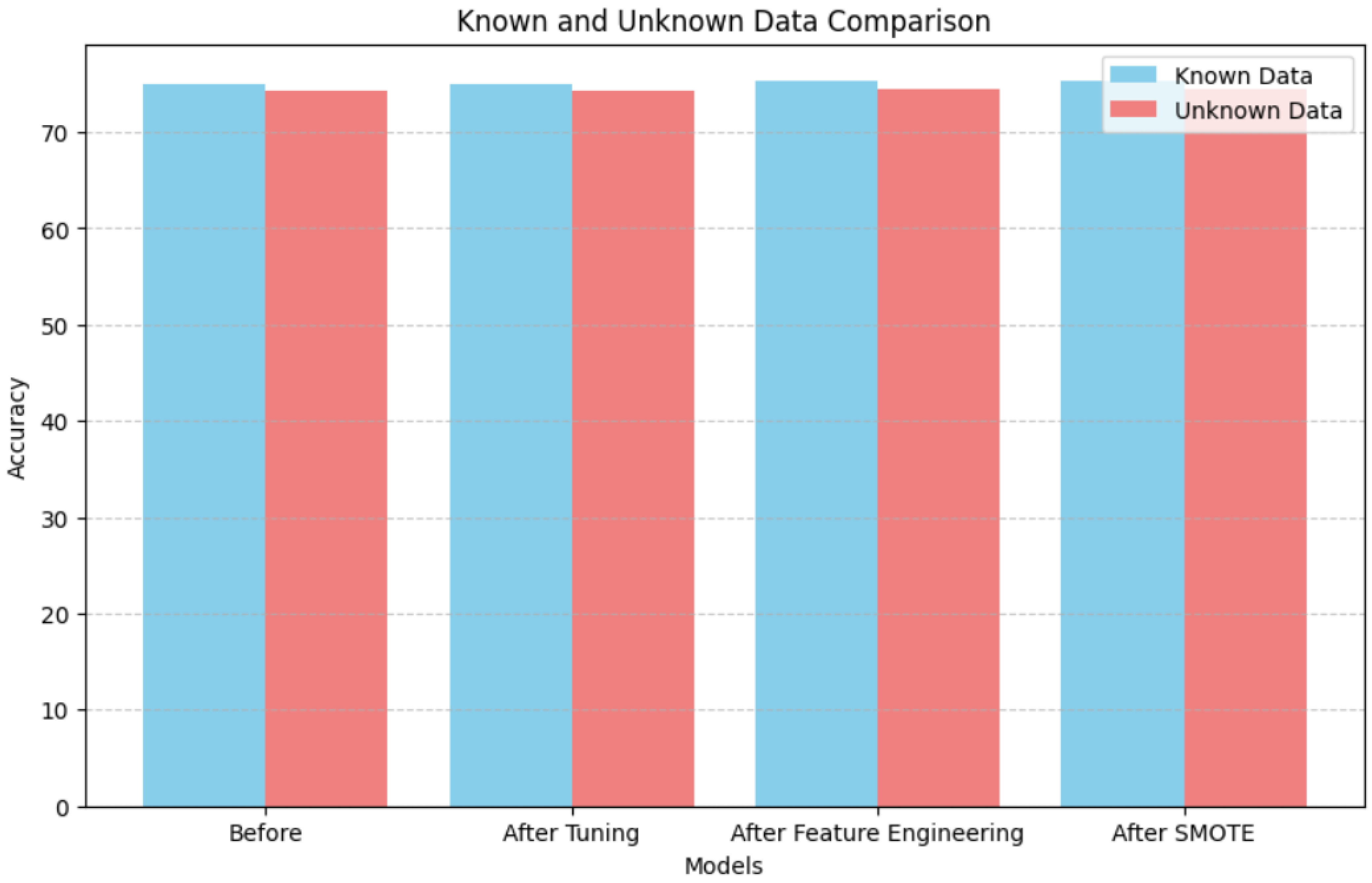

Stacked Bar Plot: AlertTitle_40 vs IncidentGrade
Count
25000
20000
15000
10000
5000
0
False
True
AlertTitle_40


Stacked Bar Plot: Category_CredentialAccess vs IncidentGrade
Count
0
5000
10000
15000
20000
False
True
Category_CredentialAccess


Stacked Bar Plot: MitreTechniques_T1552 vs IncidentGrade
Count
0
5000
10000
15000
20000
25000
0
1
MitreTechniques_T1552

Journey of model

	Known data				Unknown data			
model	Accuracy	F1 Score	Recall	Precision	Accuracy	F1 Score	Recall	Precision
At first	74.91%	75	74	76	74.3%	74	73	75
After Tuning	74.92%	74	74	76	74.28%	74	73	75
After Feature Engineering	75.38%	75	74	75	74.5%	74	73	73
After SMOTE	75.34%	75	74	75	74.5%	74	74	74


Known and Unknown Data Comparison
Known Data
Unknown Data
Accuracy
70
60
50
40
30
20
10
0
Before
After Tuning
After Feature Engineering
After SMOTE
Models

Classification Report:

class	f1-score
0	0.742348
1	0.755465
2	0.738612

Accuracy on unknown data:
0.745073

Thank You