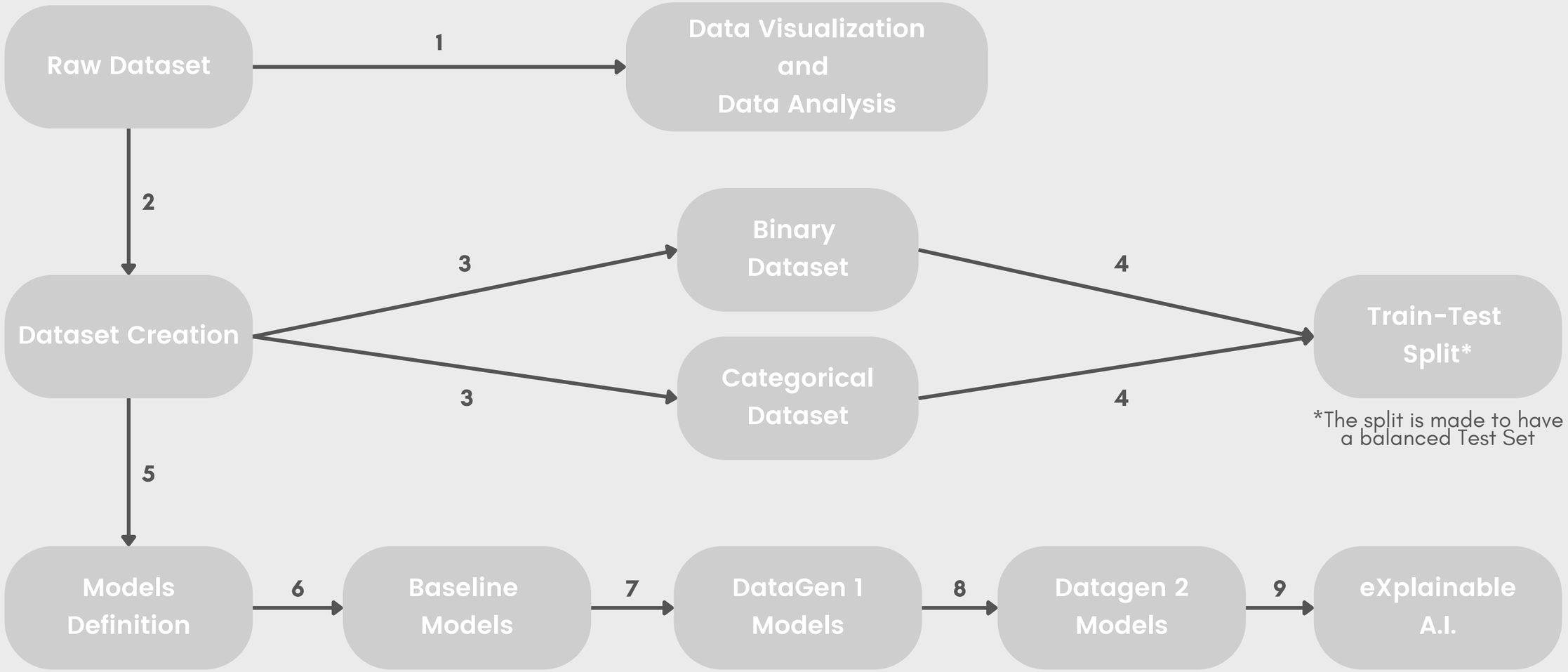
Network Measurement and Data Analysis Lab

Project 10 – The more, the merrier?!

A.A. 2023 / 2024

Guide Lines


Raw Dataset
1
Data Visualization and Data Analysis
2
Dataset Creation
3
Binary Dataset
3
Categorical Dataset
4
4
Train-Test Split*
*The split is made to have a balanced Test Set
5
Models Definition
6
Baseline Models
7
DataGen 1 Models
8
Datagen 2 Models
9
eXplainable A.I.

The Task

In the context of microwave networks, classifying the following types of errors:

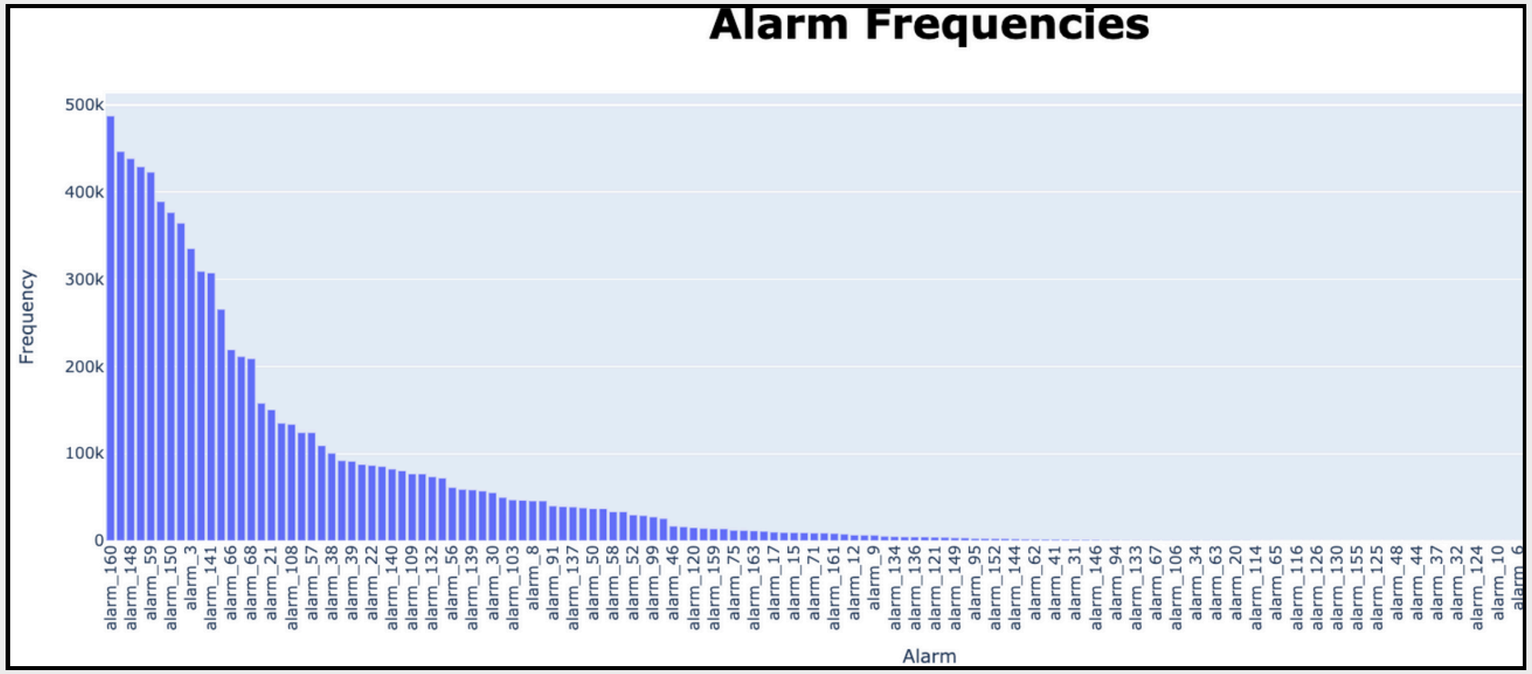
- In-Door Unit Failure (**Class 0**)
  - ↳ 515 Samples
- Out-Door Unit Failure (**Class 1**)
  - ↳ 611 Samples
- Cable Failure (**Class 2**)
  - ↳ 207 Samples
- Power Failure (**Class 3**)
  - ↳ 336 Samples

	alarm_0	alarm_1	alarm_2	alarm_3	alarm_4	alarm_6	alarm_7	alarm_9	alarm_10
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...
1664	0	0	0	0	0	0	0	0	0
1665	0	0	0	0	0	0	0	0	307
1666	0	0	0	0	0	0	0	0	1
1667	0	0	0	0	0	0	0	0	0
1668	0	0	0	0	0	0	0	0	1

and evaluating the impact of SMOTE and GAN augmentation on the model’s performance

The dataset is related to the following paper: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10543089

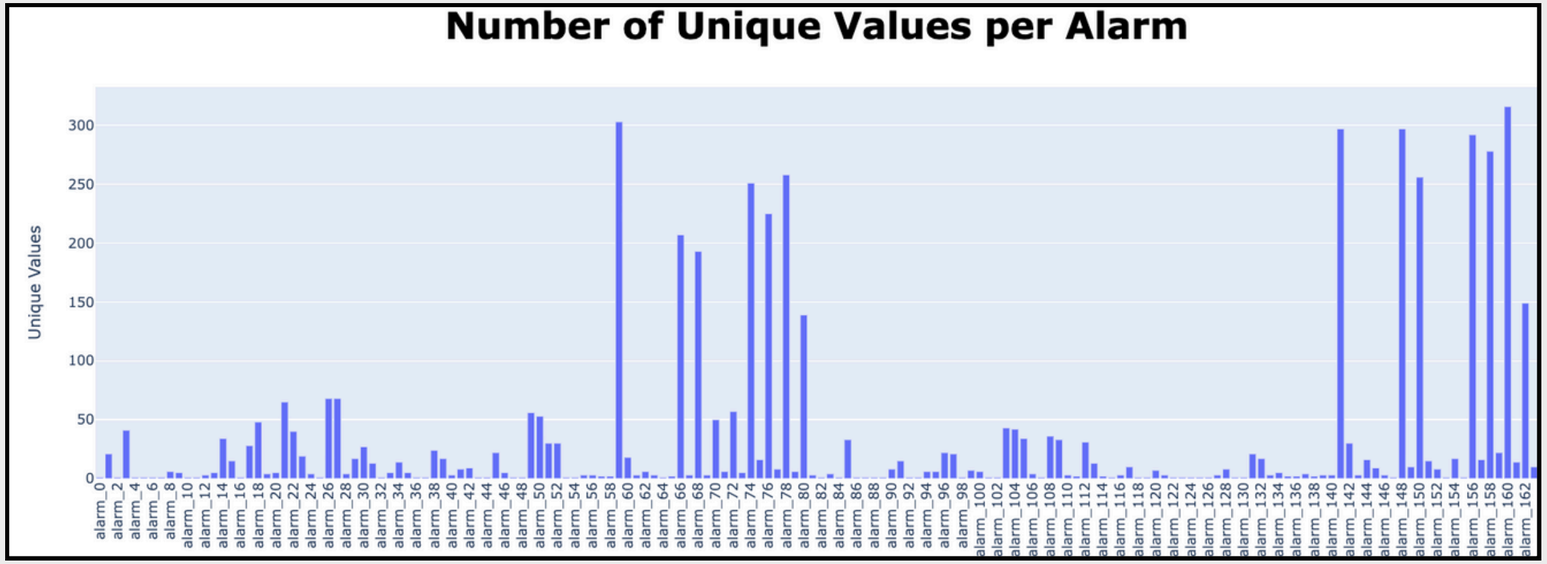
Visualization and Analysis

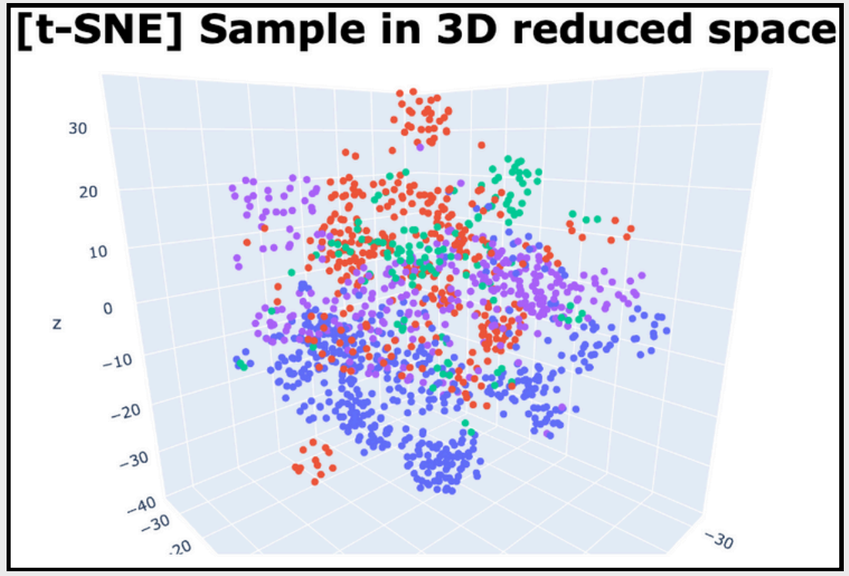


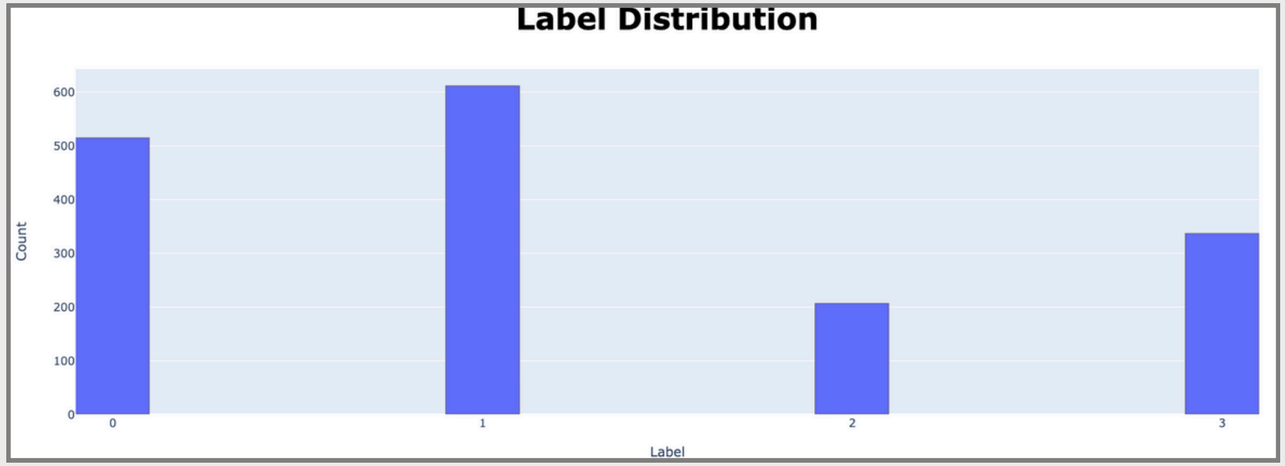
Can we extract something?

- Missing Values? No
- Duplicates? 534
- Out-of-Bound Values? No
- Constant Columns? 45

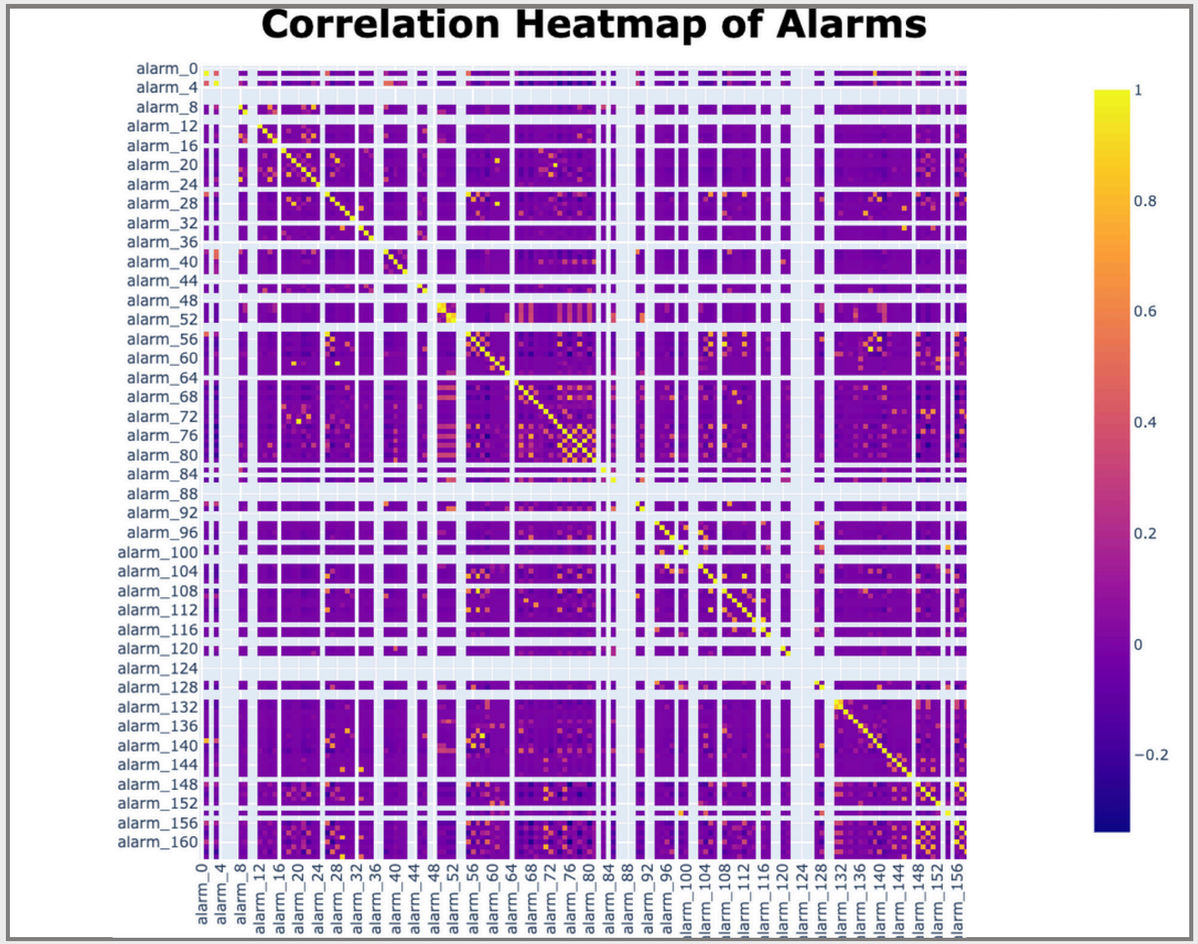
Visualization and Analysis

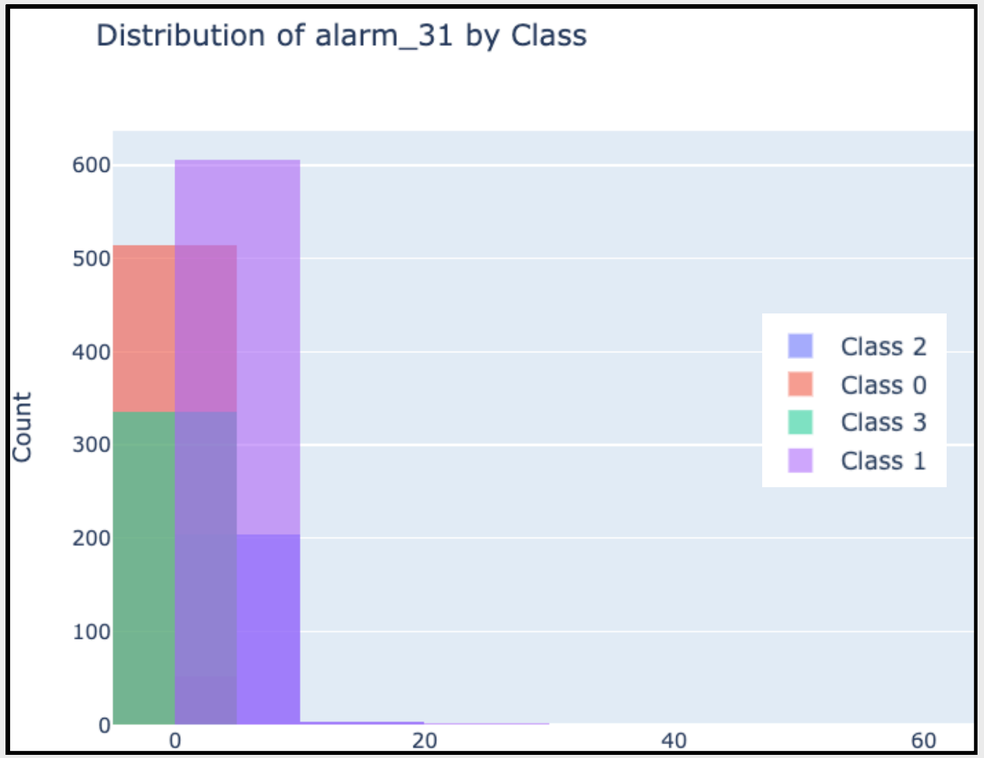






Visualization and Analysis





Dataset Creation

Data Shapes

	xTrain	yTrain	xTest	yTest
Binary	(1169, 119)	(1169,)	(500, 119)	(500,)
Categorical	(1169, 119)	(1169,)	(500, 119)	(500,)

Data Distribution

	Class 0	Class 1	Class 2	Class 3
Training	390	486	82	211
Test	125	125	125	125

The dataset has been modified by removing the constant columns and splitting in a balanced manner, where the split has the ratio 70/30 %

DataGen

Class 0	Class 1	Class 2	Class 3
390	486	82	211

Original Sample Distibution per class

Class 0	Class 1	Class 2	Class 3
486	486	486	486

DataGen1: augment each class to the number of the majority one

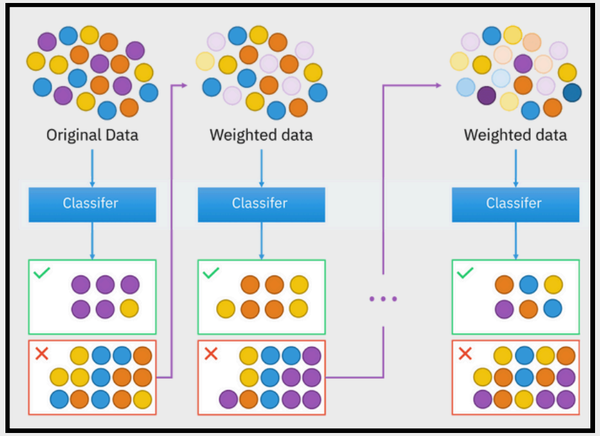
Class 0	Class 1	Class 2	Class 3
390	486	17	211

DataGen2/A: remove a 80% of samples from a class (see table) and rebalance as DataGen 1

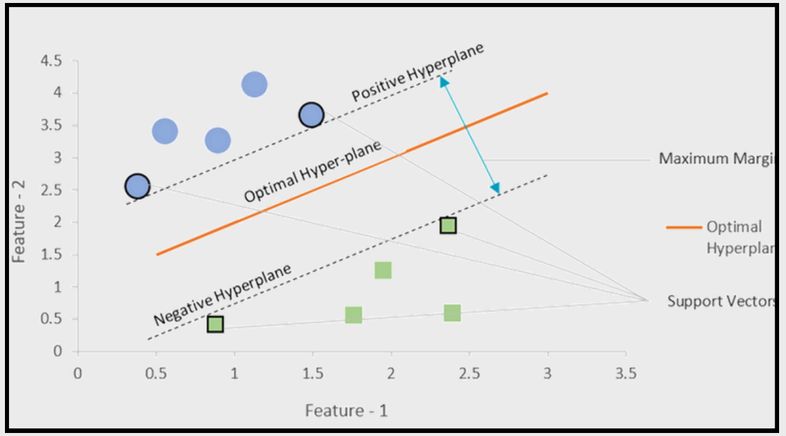
Class 0	Class 1	Class 2	Class 3
690	786	317	511

DataGen2/B: remove a 80% of samples from a class (see table) and augment each class with 300 samples

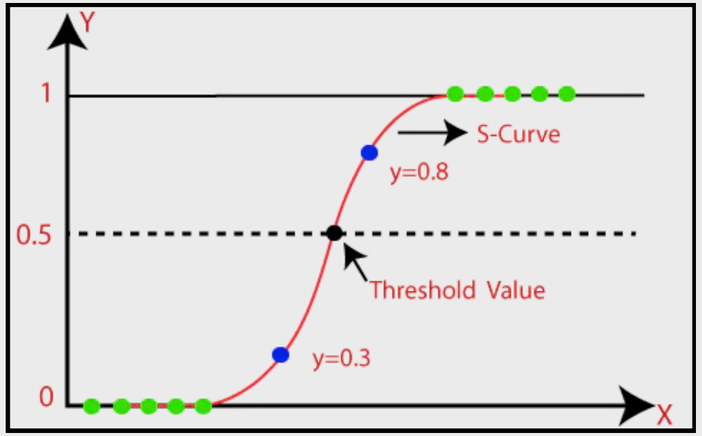
Models Definition



XGBoost [1] is an optimized gradient boosting machine learning algorithm designed for speed and performance



Support Vector Machine (SVM) [2] is a powerful supervised learning algorithm used mostly for classification, which works by finding the optimal hyperplane that best separates different classes in the feature space



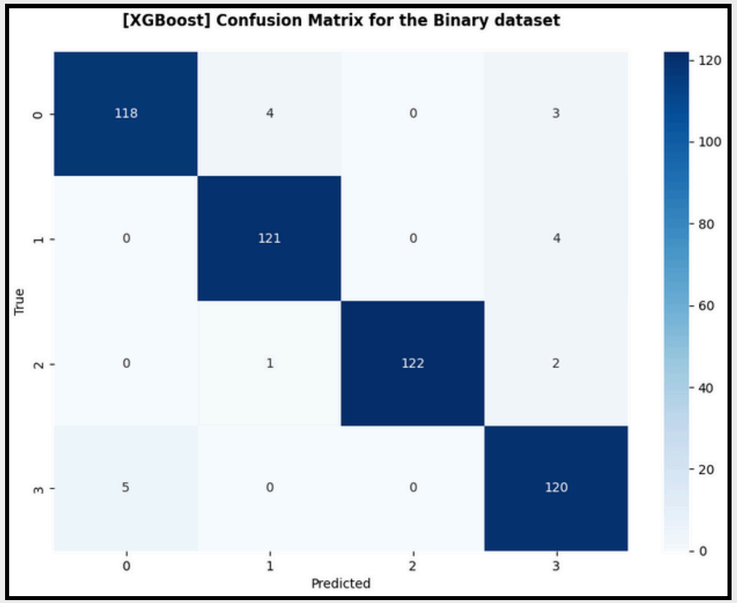
Logistic Regression [3] is a statistical method that models the probability of a binary outcome using a logistic function

[1] https://xgboost.readthedocs.io/en/stable/
[2] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html
[3] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

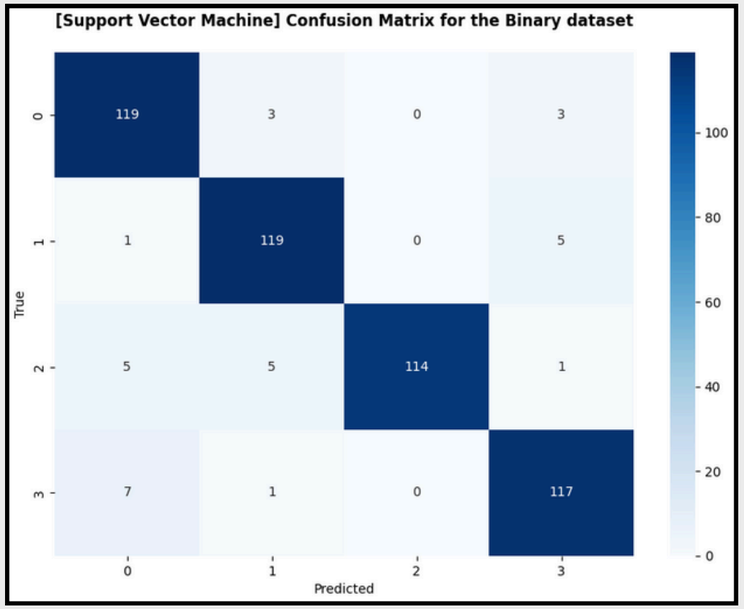
Baseline Models - DataGen 1

Binary

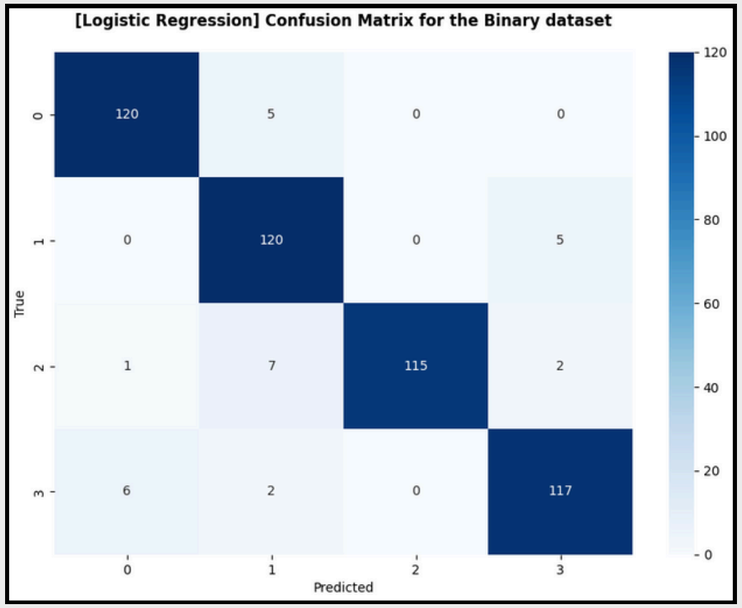
XGBoost



Support Vector Machine



Logistic Regression



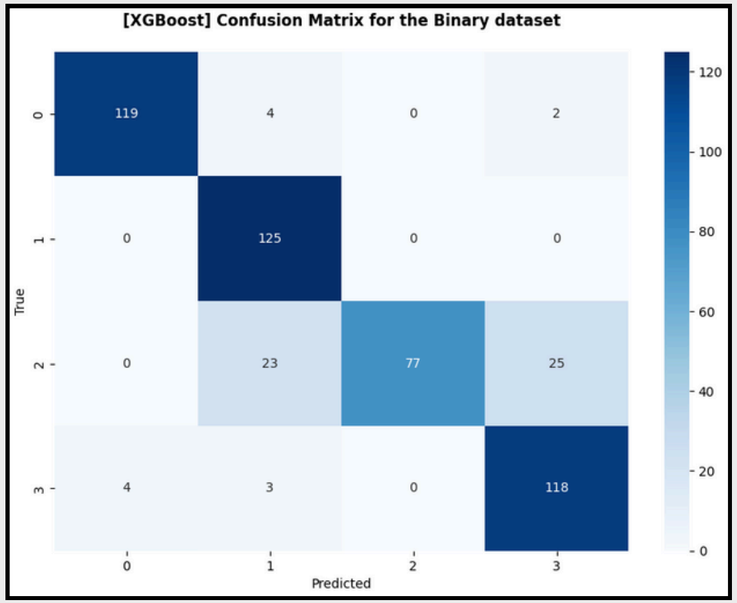
The hyper-parameters has been found with Stratified K-Fold Cross-Validation (K = 10):

- **XGBoost:** # Estimators = 100, Max Depth = 5, Subsample = 0.8, Learning Rate = 0.2 (5 min)
- **SVM:** Decision Function = O-v-O, Kernel = linear, Gamma = Scale, C = 1 (2 min)
- **LR:** Solver = Sag, Max Iteration = 1000, C = 10 (6 min)

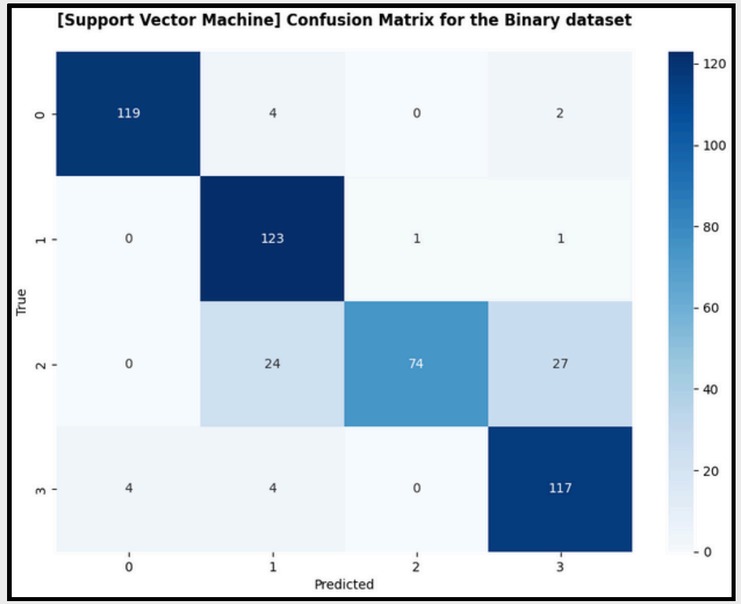
Baseline Models - DataGen 2

Binary

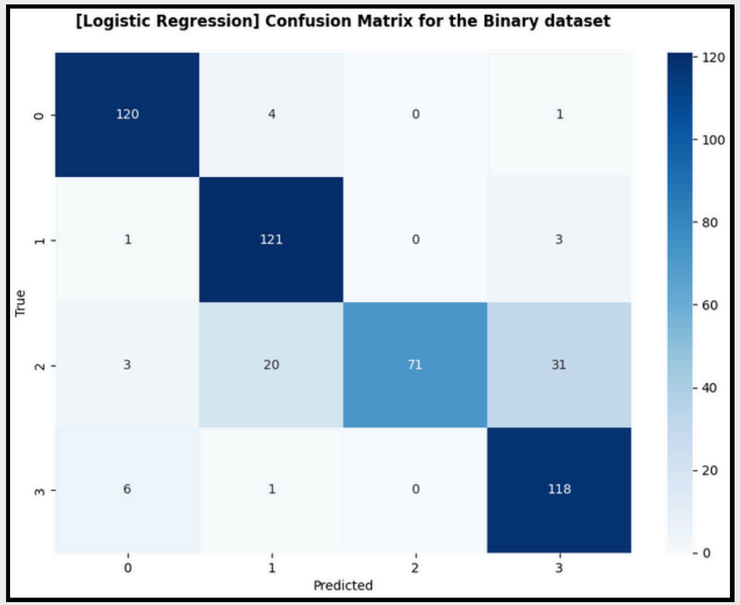
XGBoost



Support Vector Machine



Logistic Regression



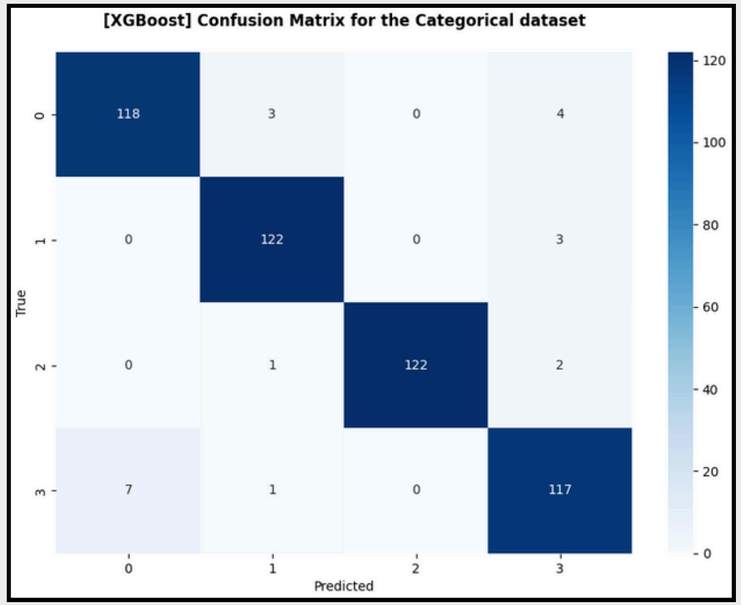
The hyper-parameters has been found with Stratified K-Fold Cross-Validation (K = 10):

- **XGBoost:** # Estimators = 100, Max Depth = 10, Subsample = 0.8, Learning Rate = 0.2 (6 min)
- **SVM:** Decision Function = O-v-O, Kernel = RBF, Gamma = Scale, C = 10 (1 min)
- **LR:** Solver = Sag, Max Iteration = 1000, C = 10 (6 min)

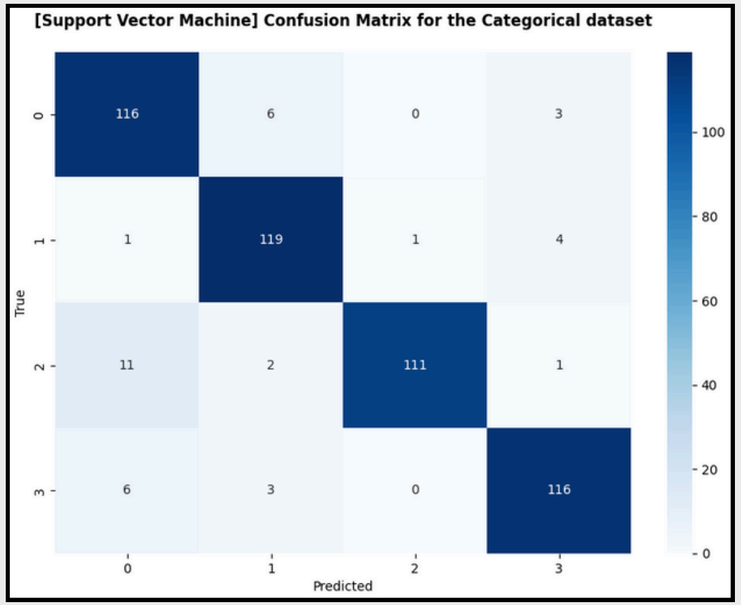
Baseline Models - DataGen 1

Categorical

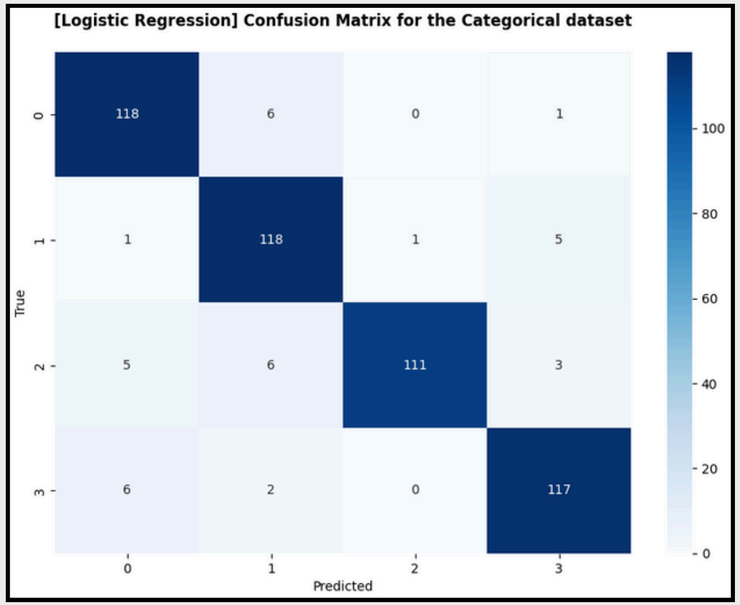
XGBoost



Support Vector Machine



Logistic Regression



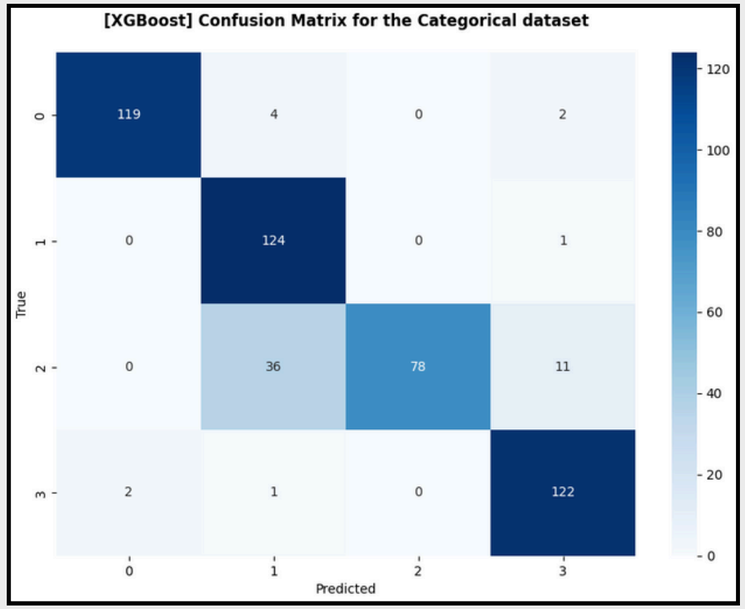
The hyper-parameters has been found with Stratified K-Fold Cross-Validation (K = 10):

- **XGBoost:** # Estimators = 150, Max Depth = 15, Subsample = 0.8, Learning Rate = 0.2 (6 min)
- **SVM:** Decision Function = O-v-O, Kernel = linear, Gamma = Scale, C = 1 (1 min)
- **LR:** Solver = Sag, Max Iteration = 5000, C = 10 (10 min)

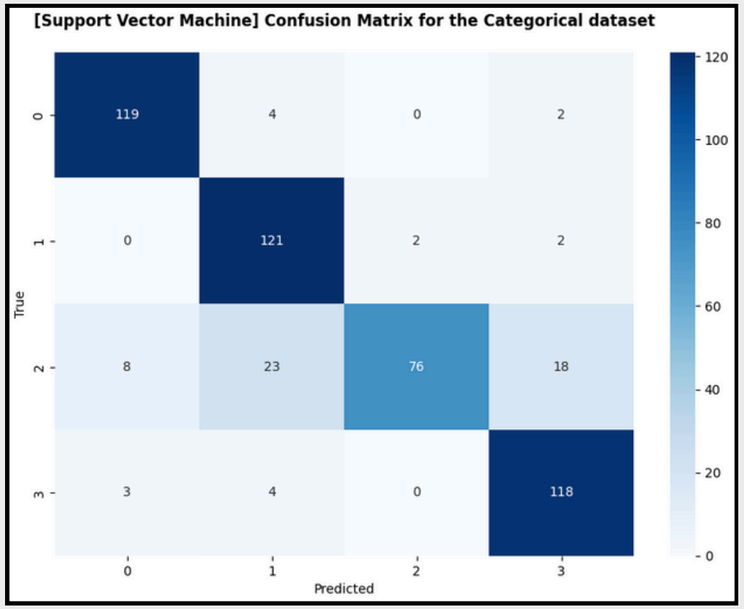
Baseline Models - DataGen 2

Categorical

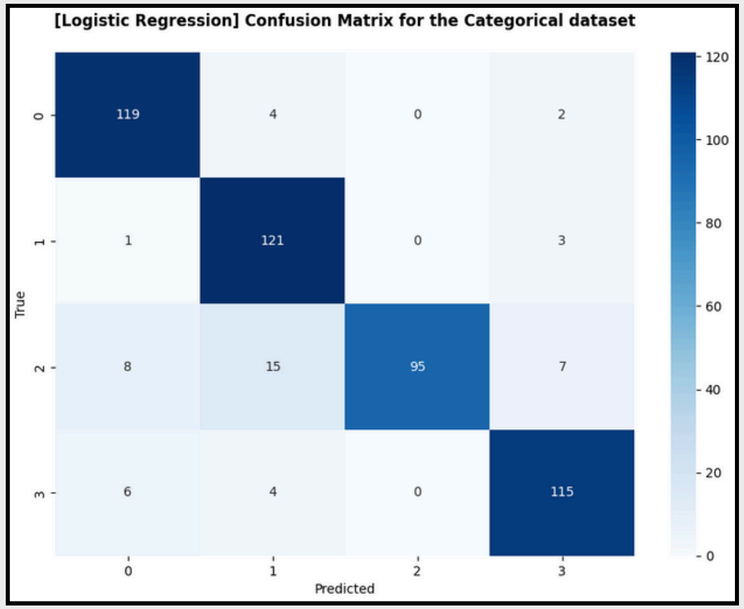
XGBoost



Support Vector Machine



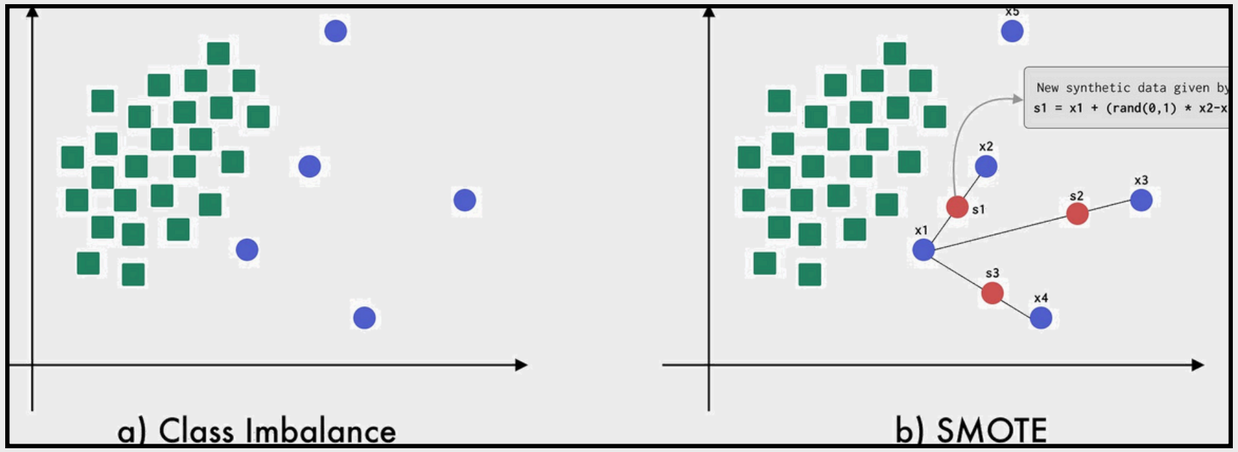
Logistic Regression

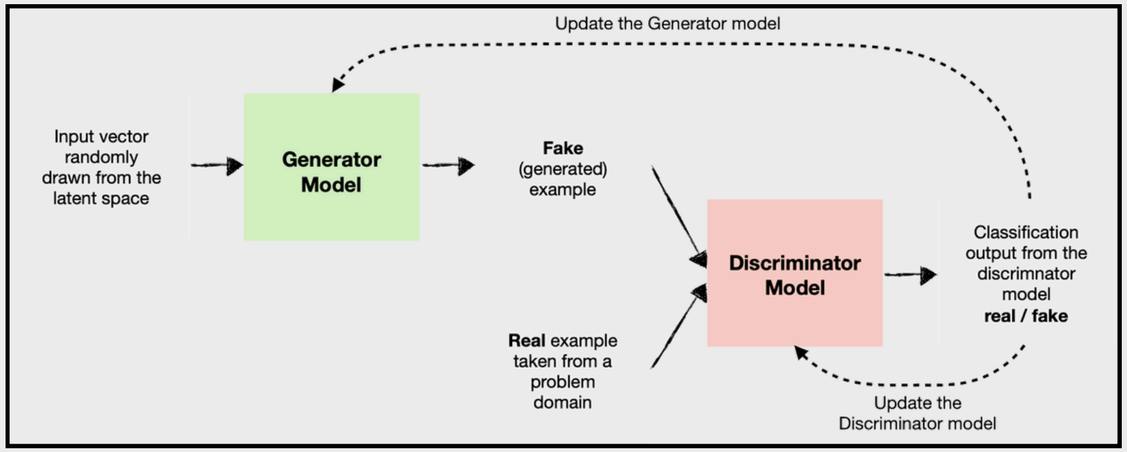


The hyper-parameters has been found with Stratified K-Fold Cross-Validation (K = 10):

- **XGBoost:** # Estimators = 150, Max Depth = 10, Subsample = 0.9, Learning Rate = 0.2 (5 min)
- **SVM:** Decision Function = O-v-O, Kernel = RBF, Gamma = Scale, C = 100 (1 min)
- **LR:** Solver = Sag, Max Iteration = 10000, C = 10 (11 min)

Augmentation Technique





SMOTE (Synthetic Minority Over-sampling Technique) [1] is a technique used to address class imbalance in datasets by generating synthetic examples of the minority class to improve model performance using the interpolation of true samples in that class

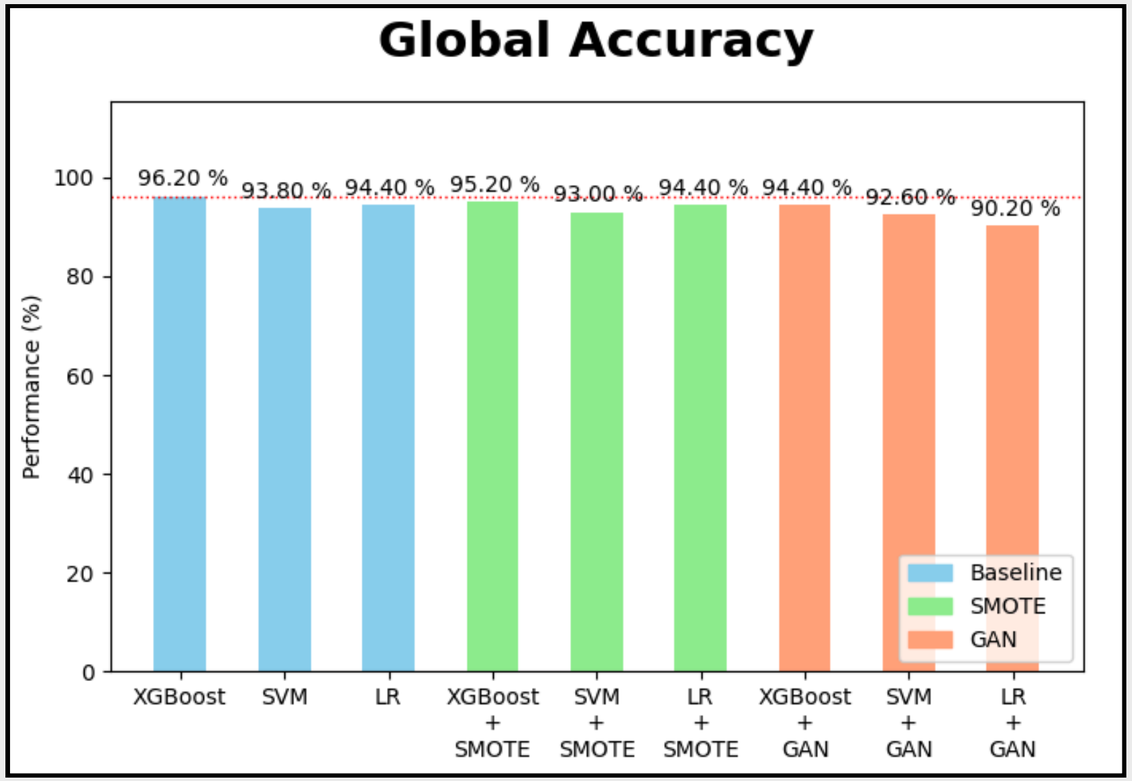
CTGAN (Conditional Tabular GAN) [2] is a deep learning model that generates realistic synthetic tabular data by learning the distribution of real data, preserving complex relationships between features for enhanced data augmentation and analysis

[1] https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html
[2] https://sdv.dev/SDV/user_guides/single_table/ctgan.html

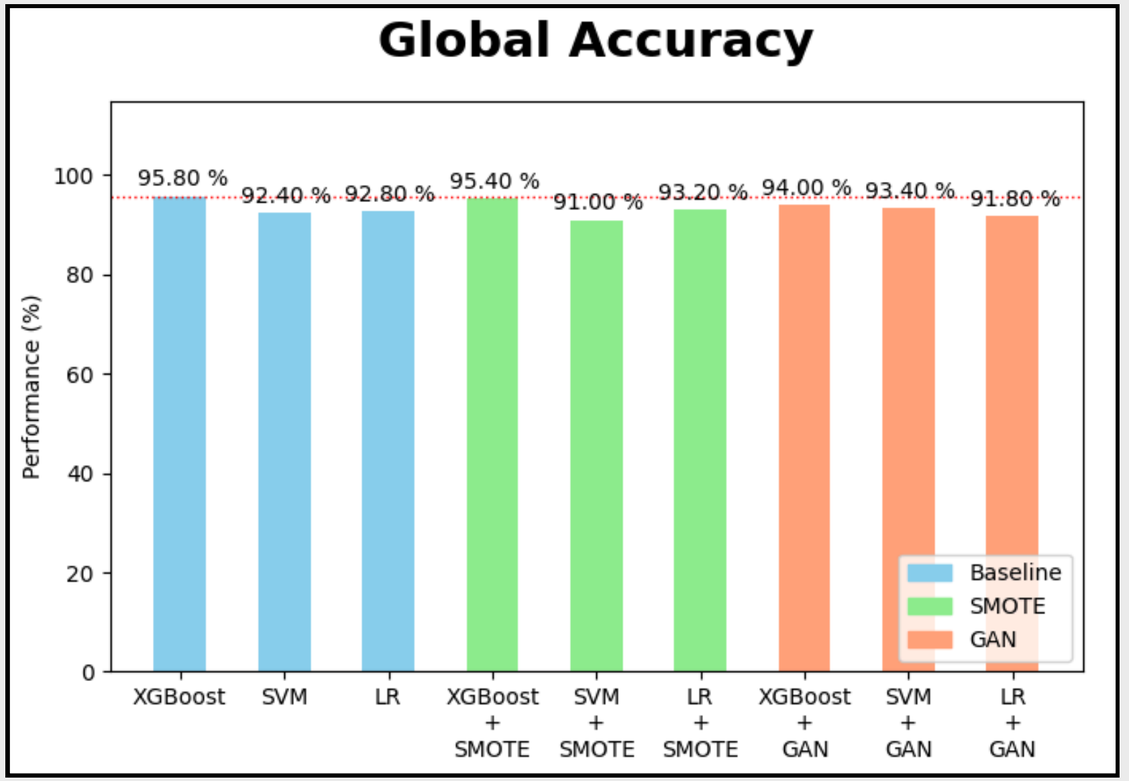
Performance Evaluation

DataGen 1

Binary



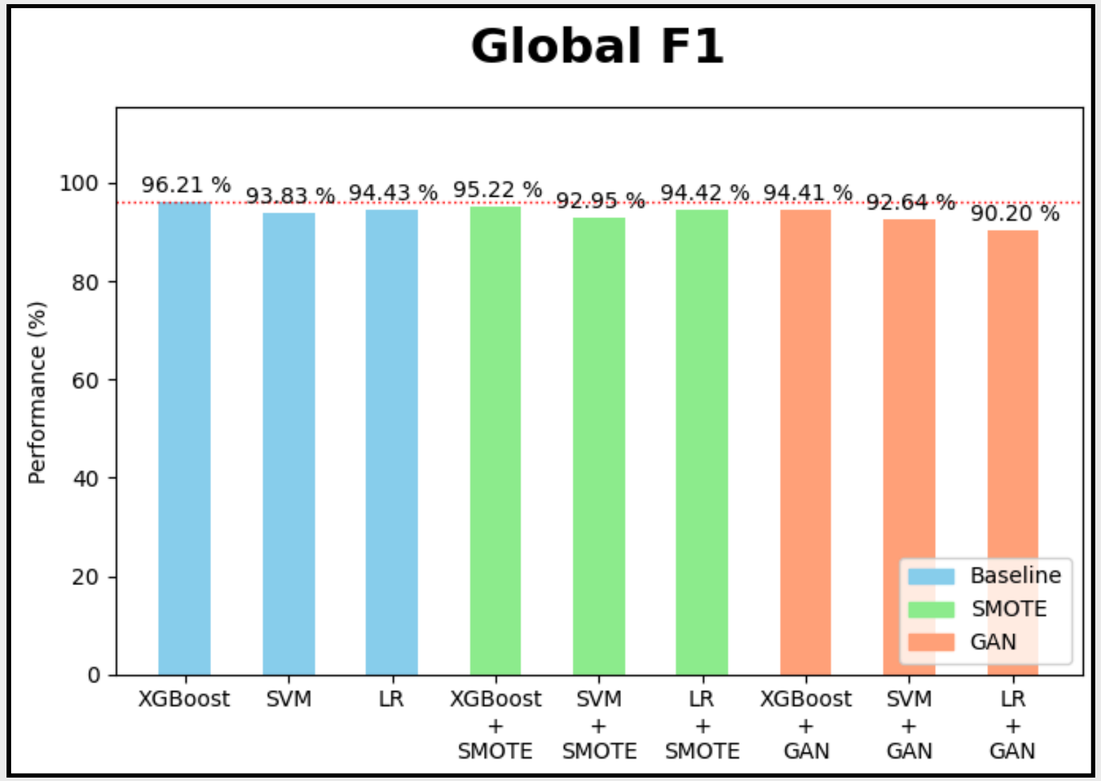
Categorical



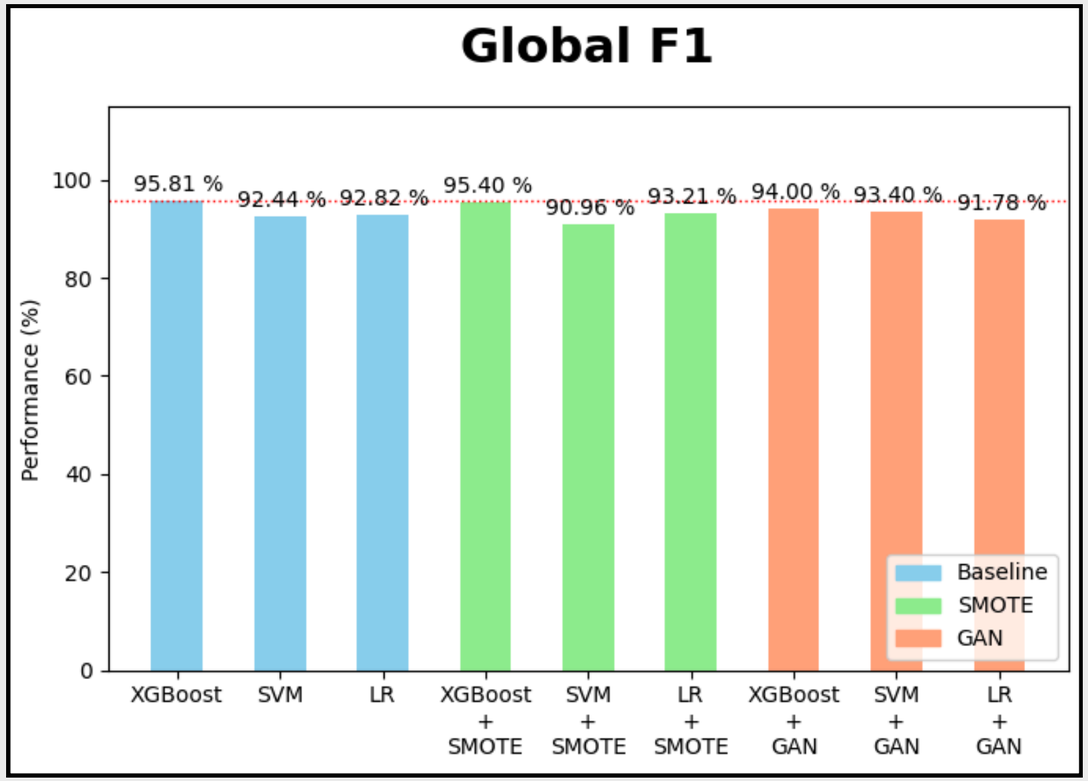
Performance Evaluation

DataGen 1

Binary



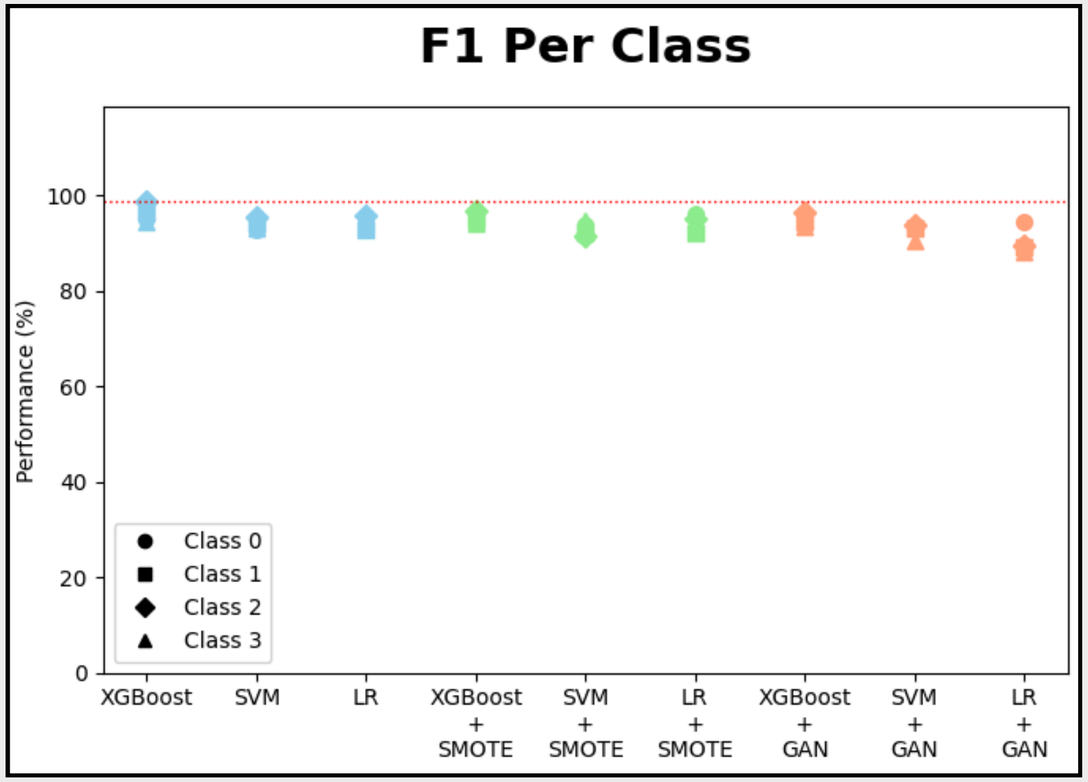
Categorical



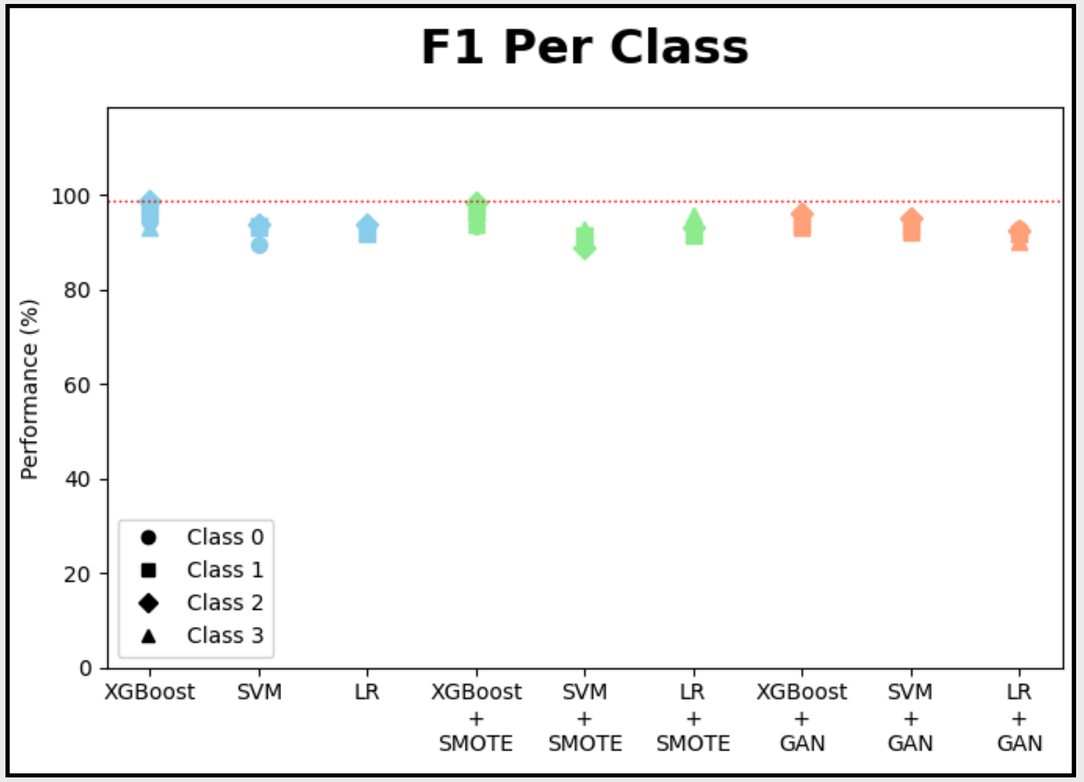
Performance Evaluation

DataGen 1

Binary



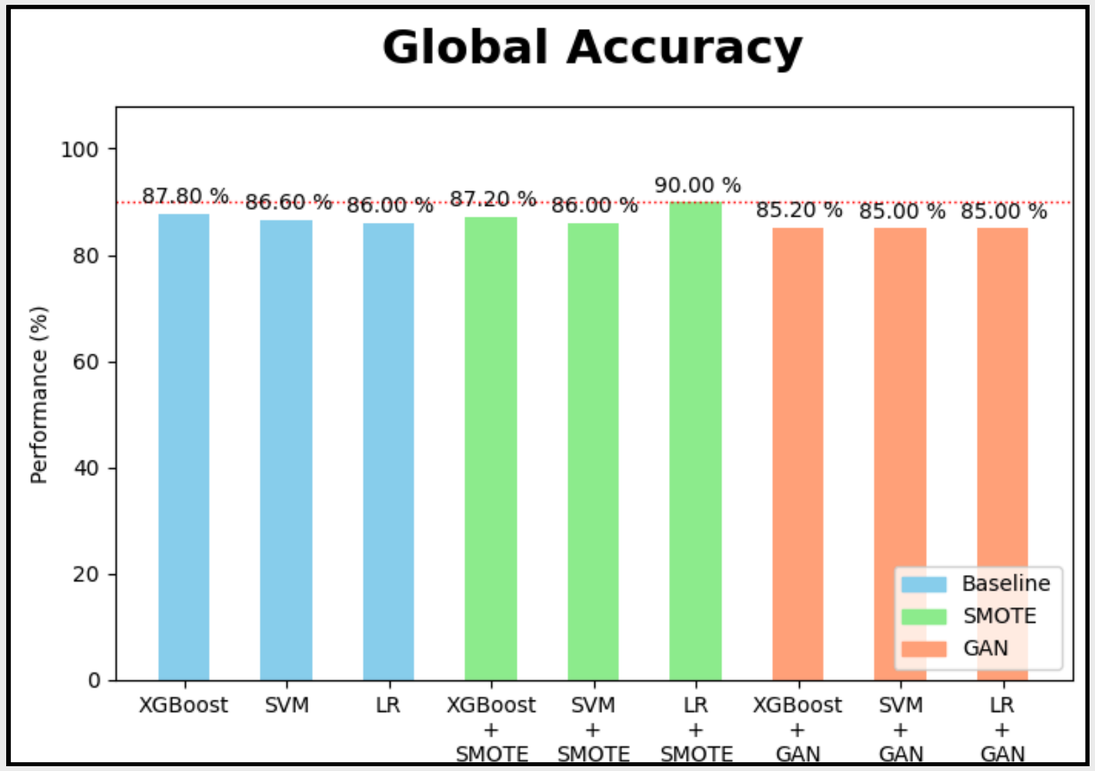
Categorical



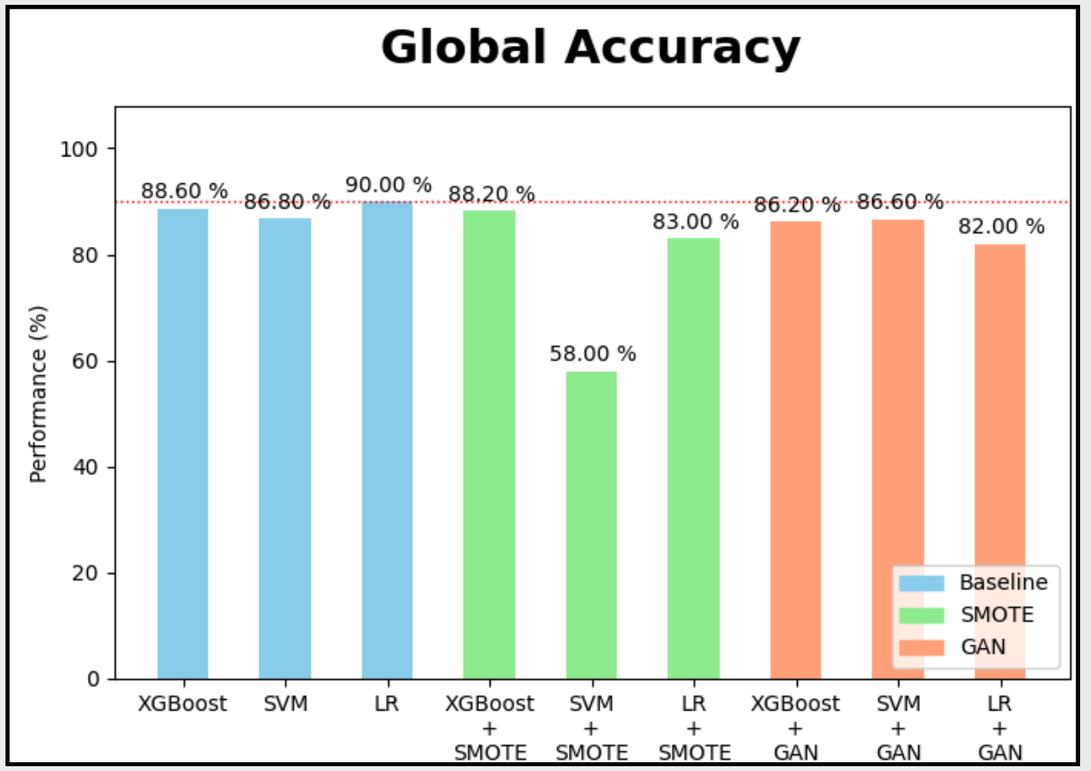
Performance Evaluation

DataGen 2/A

Binary



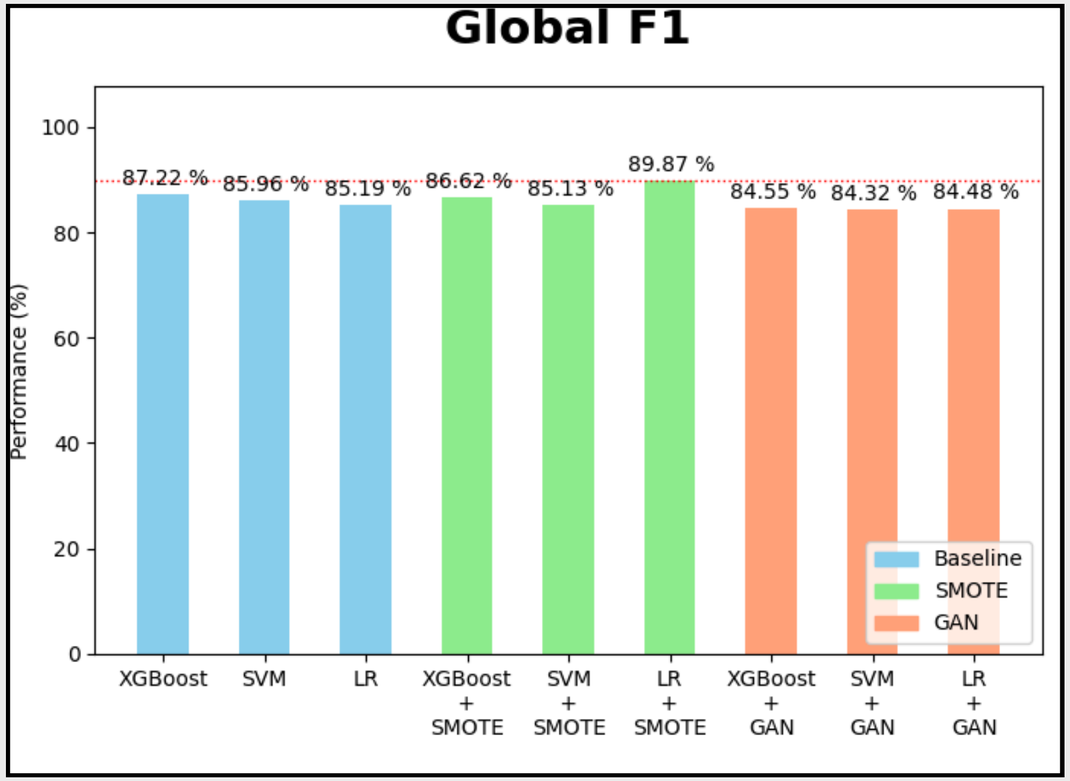
Categorical



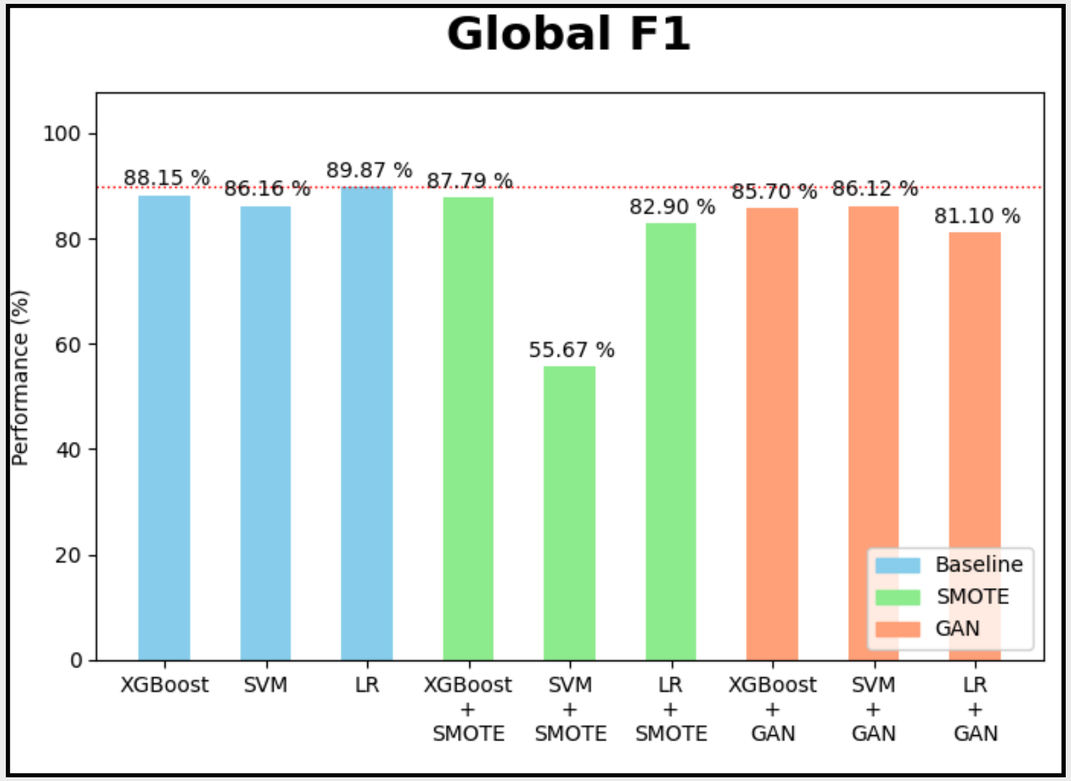
Performance Evaluation

DataGen 2/A

Binary



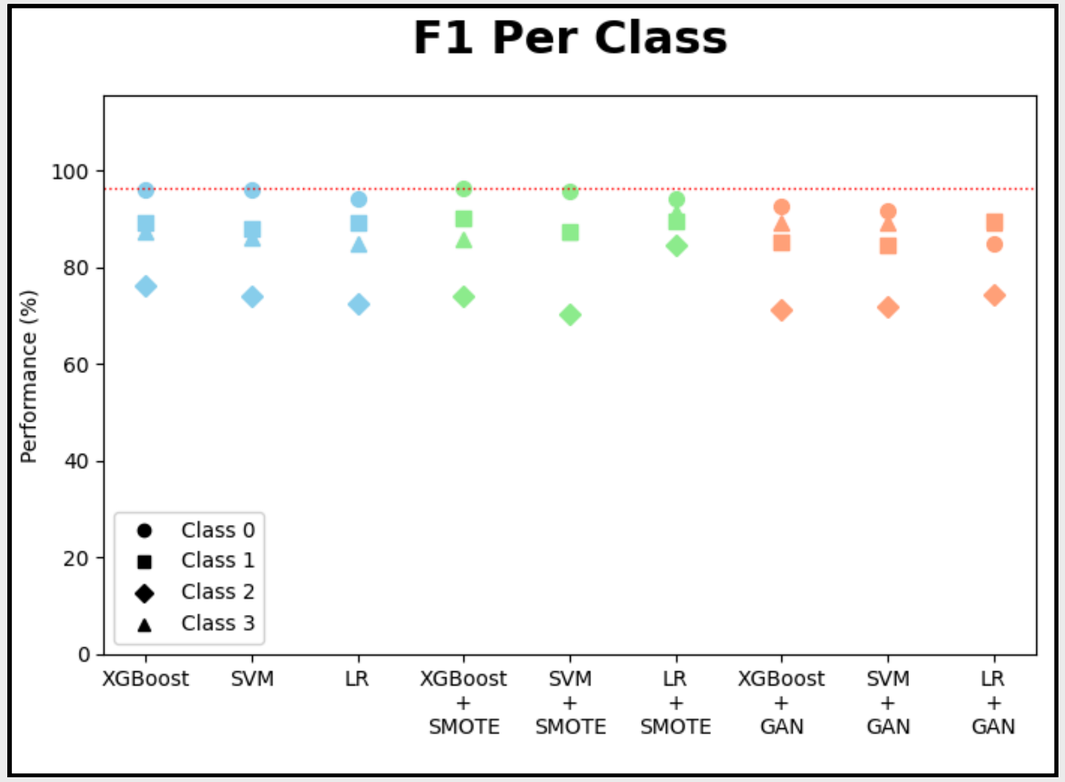
Categorical



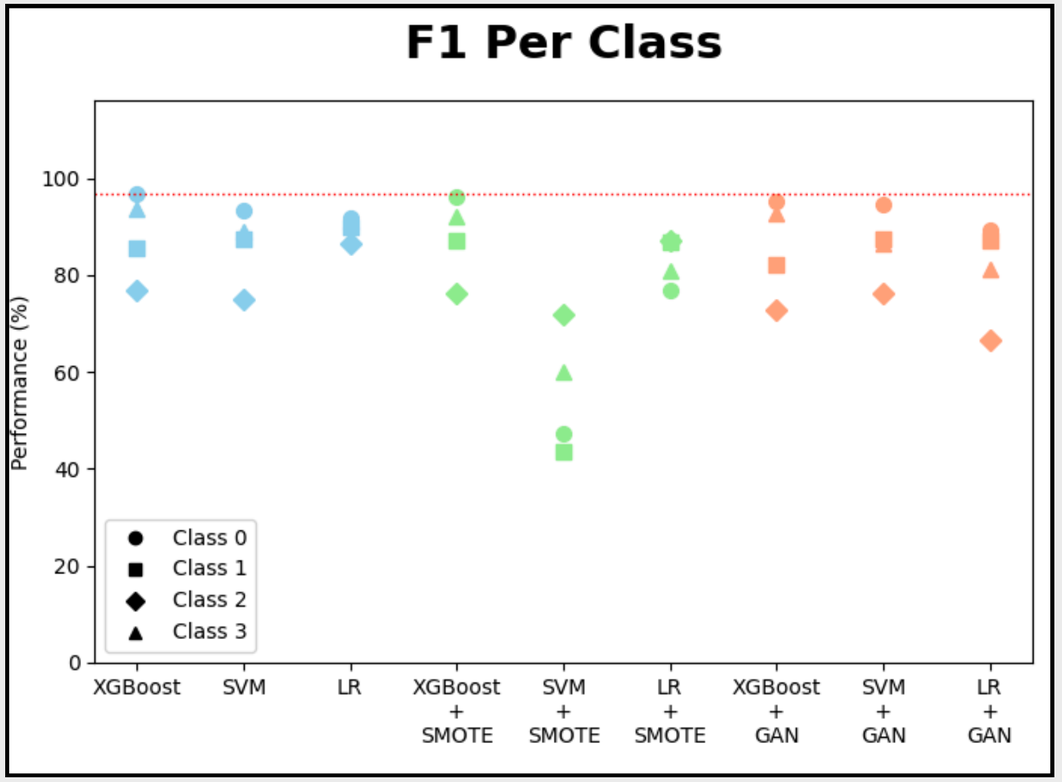
Performance Evaluation

DataGen 2/A

Binary



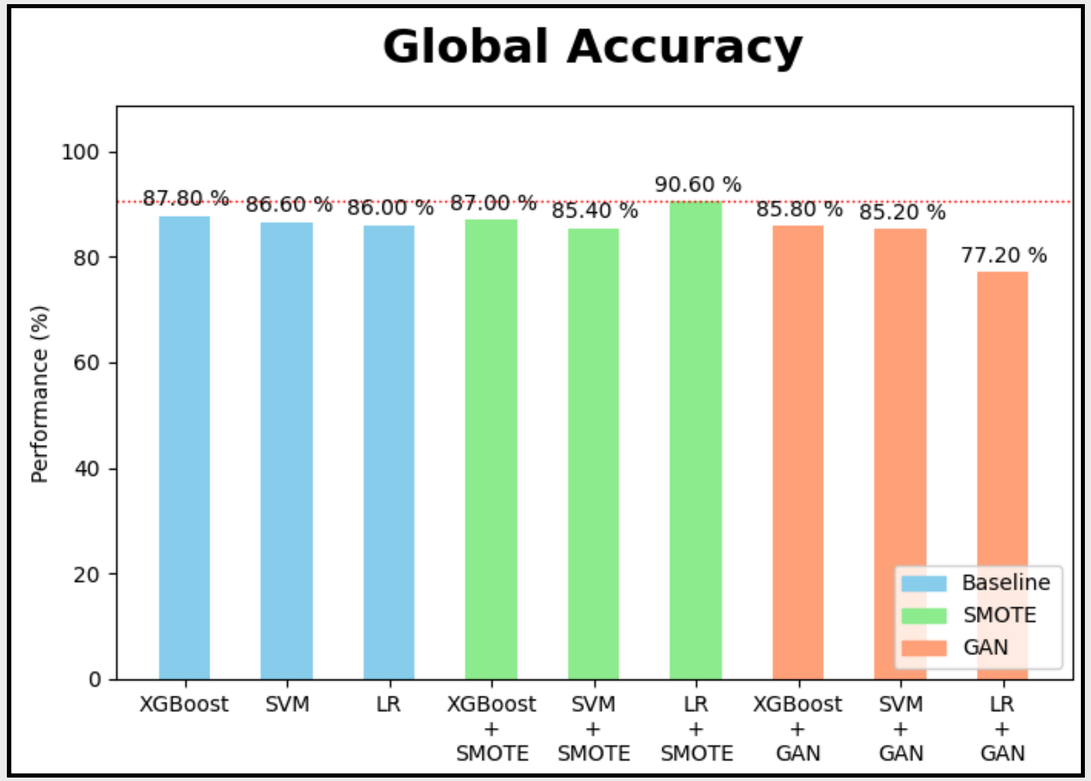
Categorical



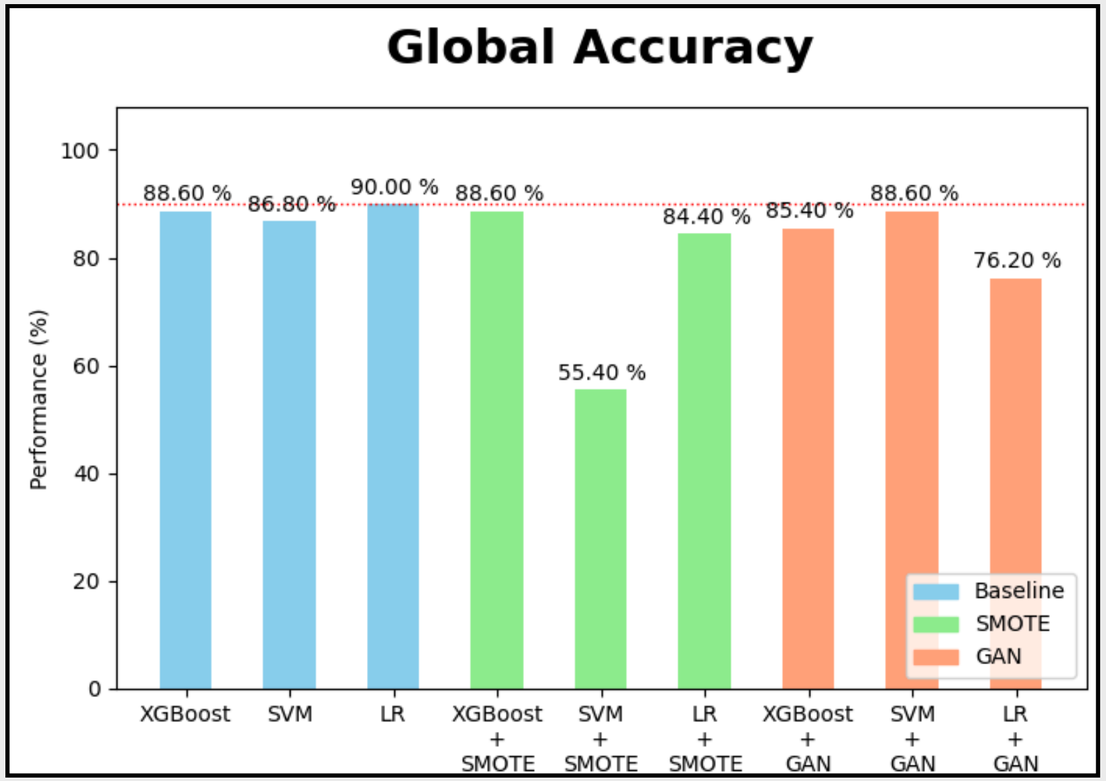
Performance Evaluation

DataGen 2/B

Binary



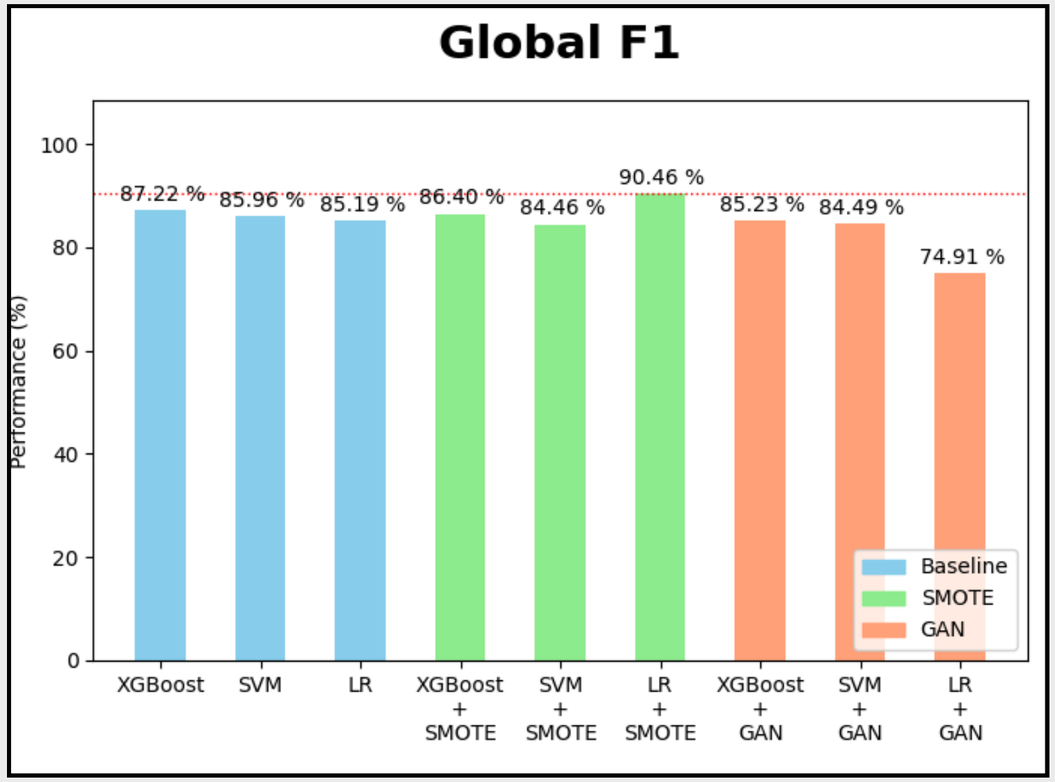
Categorical



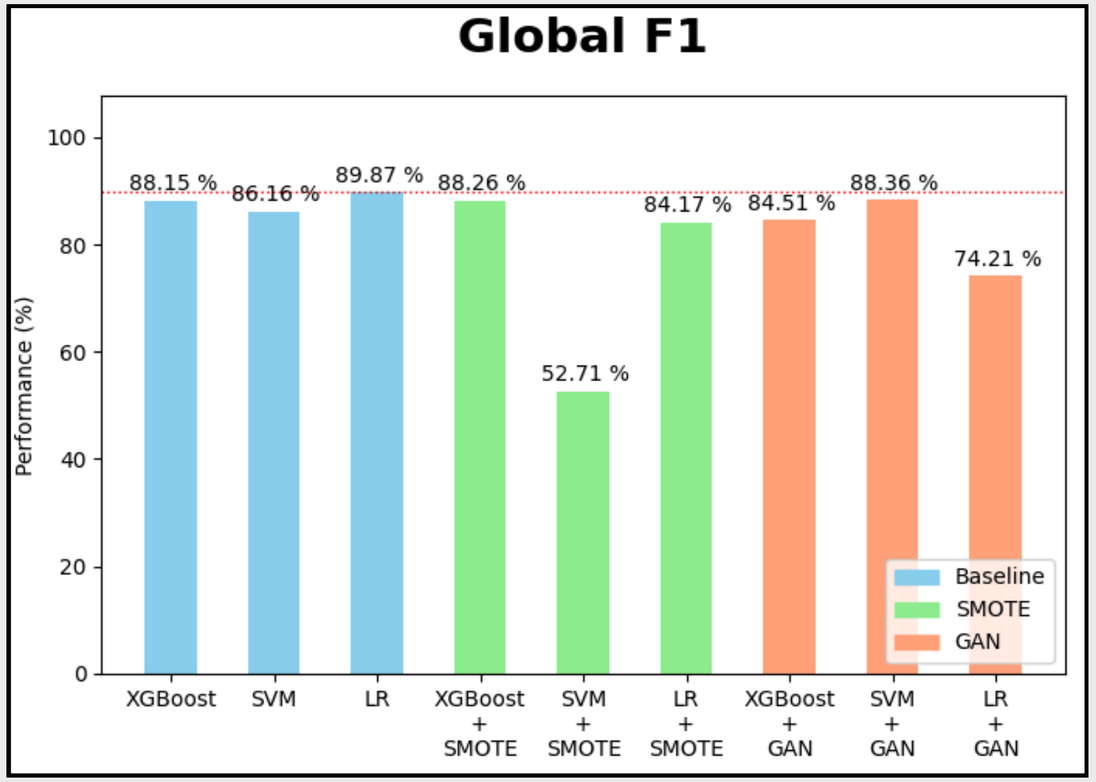
Performance Evaluation

DataGen 2/B

Binary



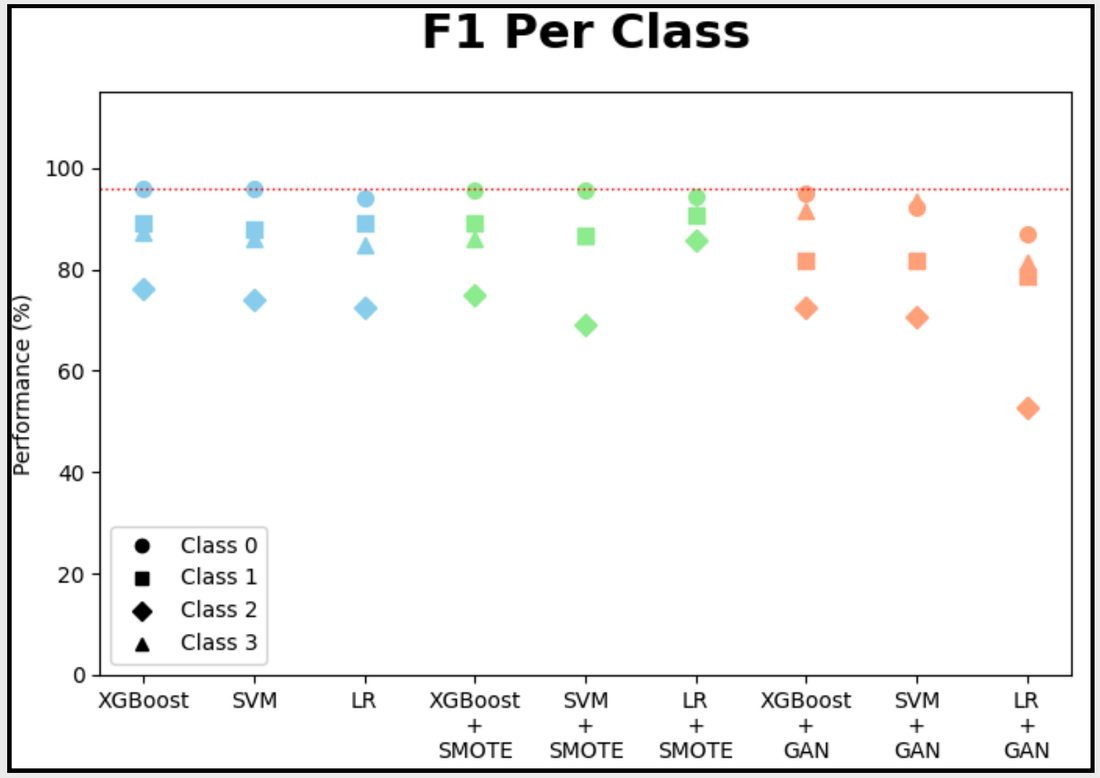
Categorical



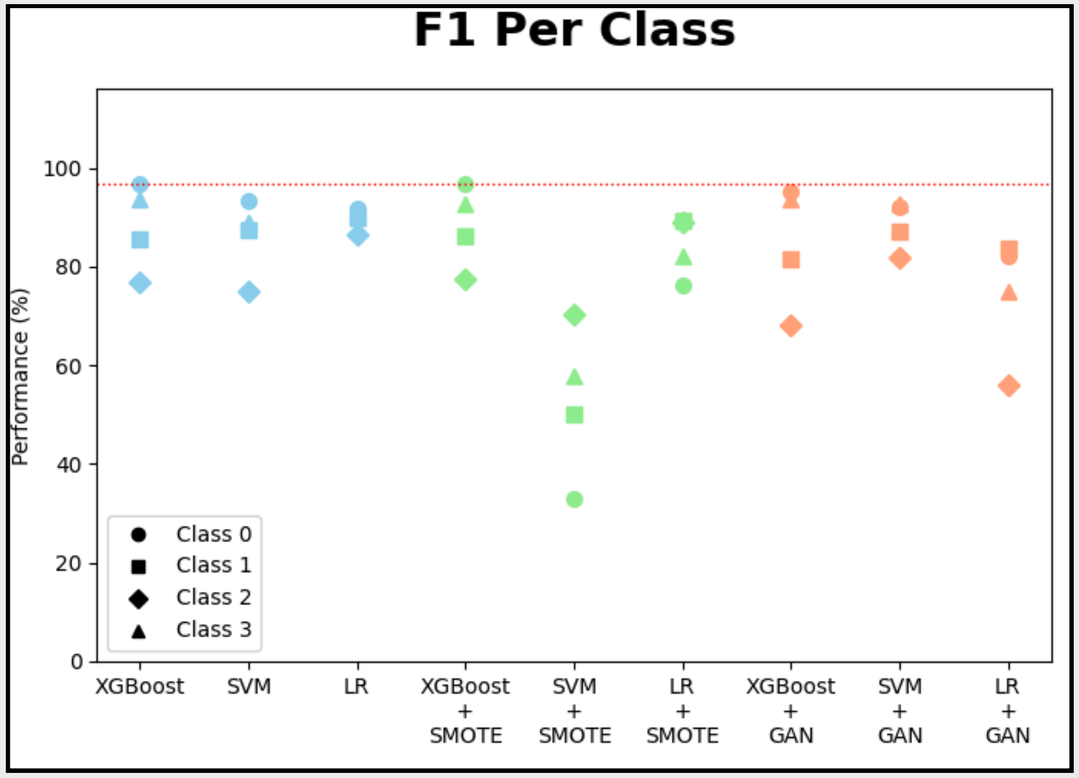
Performance Evaluation

DataGen 2/B

Binary



Categorical



Performance Evaluation

Binary

	XGBoost	SVM	LR
Baseline	[A] 0.25 s [B] 0.26 s	[A] 0.31 s [B] 0.18 s	[A] 0.68 s [B] 0.53
DataGen 1	[1] 0.46 s [2] 0.40 s	[1] 0.39 s [2] 1.19 s	[1] 1.28 s [2] 1.46 s
DataGen 2/A	[1] 1.71 s [2] 0.69 s	[1] 0.75 s [2] 1.61 s	[1] 1.98 s [2] 1.13 s
DataGen 2/B	[1] 0.58 s [2] 0.82 s	[1] 0.58 s [2] 4.07 s	[1] 1.50 s [2] 2.01 s

[1] = SMOTE Augmentation
[2] = CTGAN Augmentation

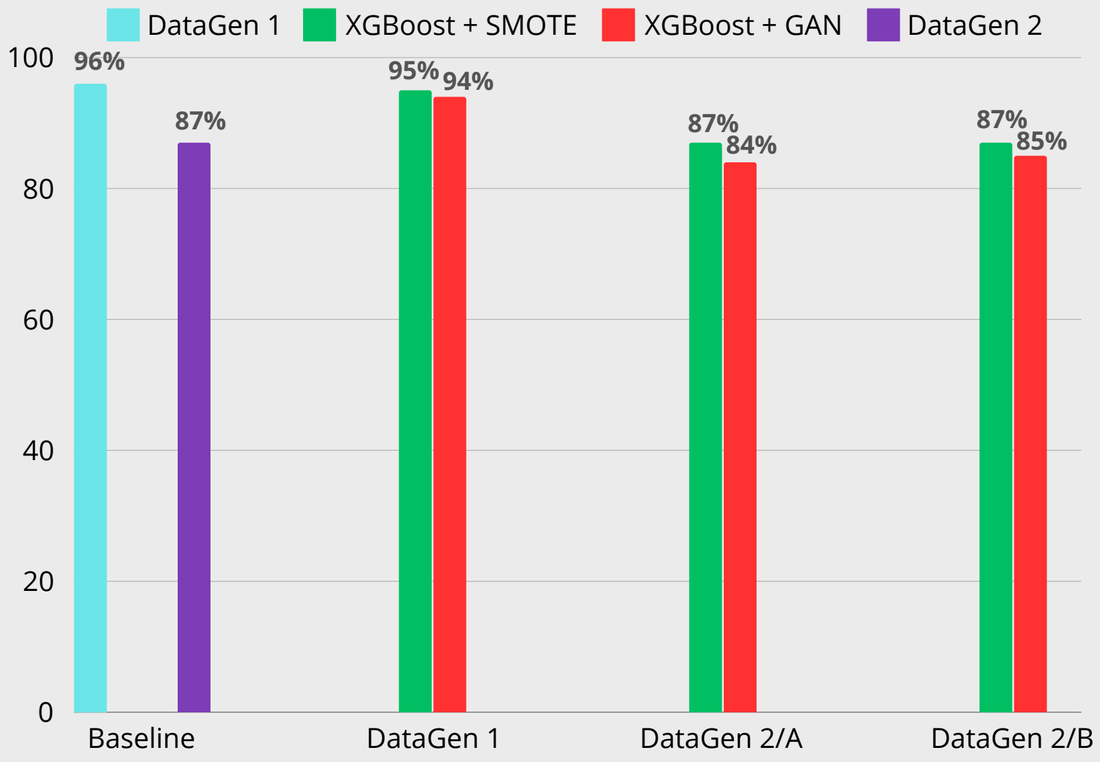
Categorical

	XGBoost	SVM	LR
Baseline	[A] 0.44 s [B] 0.38	[A] 0.19 s [B] 0.15 s	[A] 2.11 s [B] 3.57 s
DataGen 1	[1] 1.98 s [2] 1.08 s	[1] 0.63 s [2] 2.44 s	[1] 3.89 s [2] 5.63 s
DataGen 2/A	[1] 3.69 s [2] 1.01 s	[1] 0.61 s [2] 1.70 s	[1] 10.59 s [2] 6 s
DataGen 2/B	[1] 0.75 s [2] 1.21 s	[1] 0.68 s [2] 4.08 s	[1] 5.62 s [2] 5.67 s

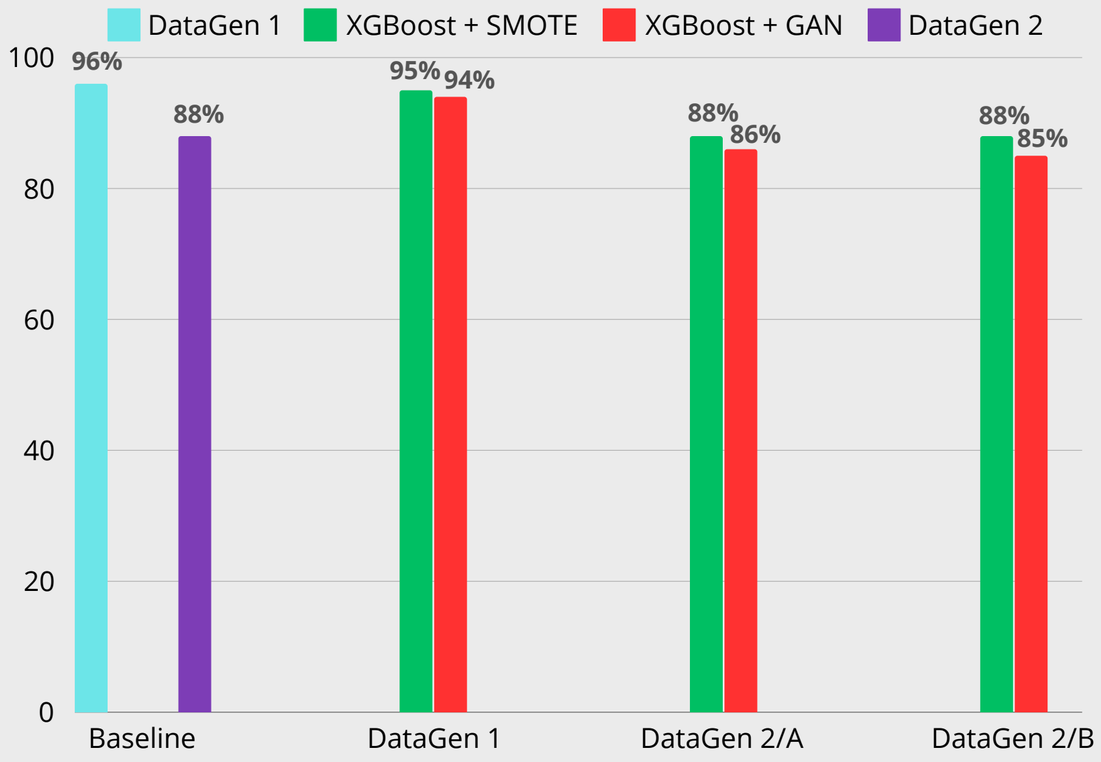
[A] = DataGen 1
[B] = DataGen 2

What is the best?

Binary

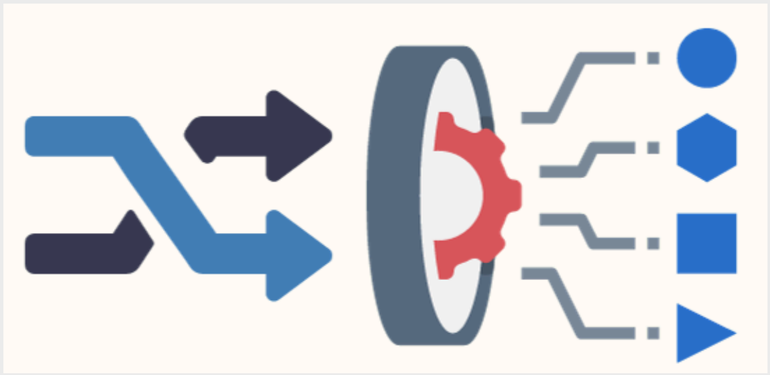


Categorical



The considered metric is the Global F1-Score (Macro)

eXplainable Artificial Intelligence

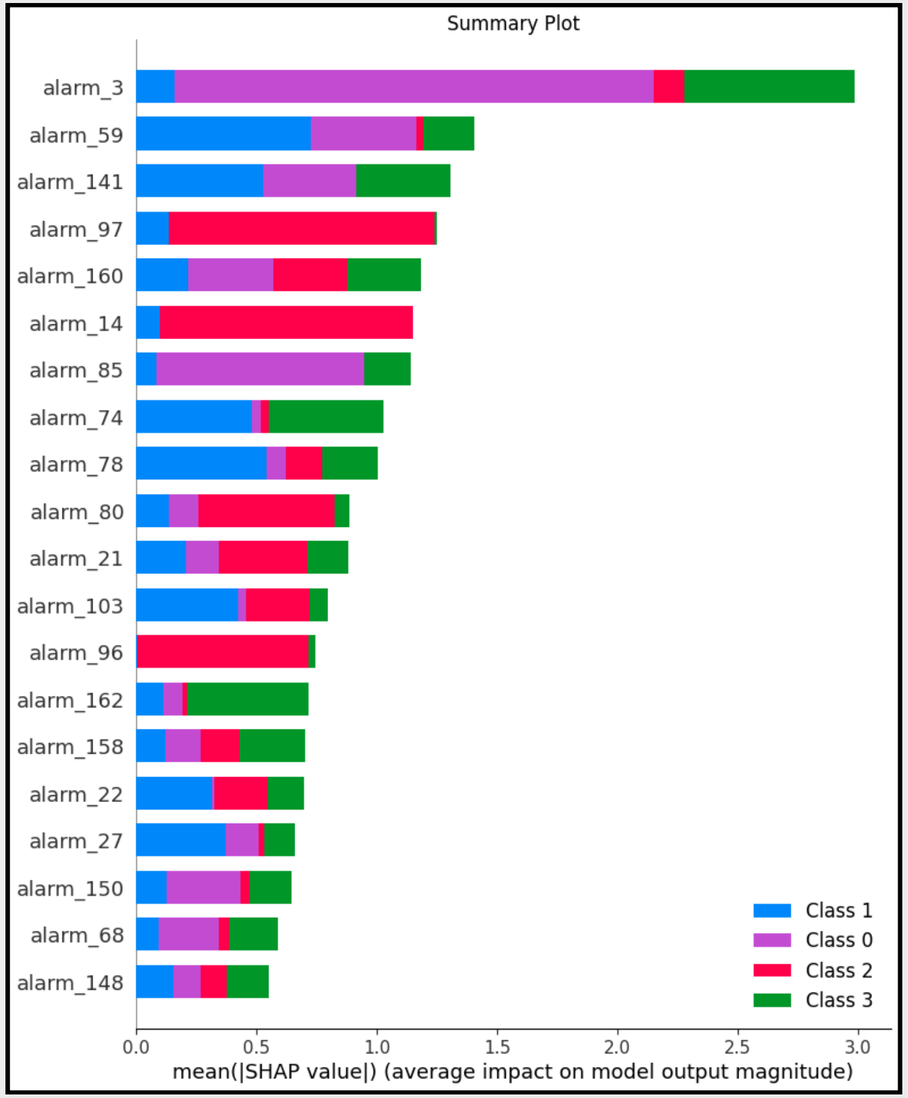


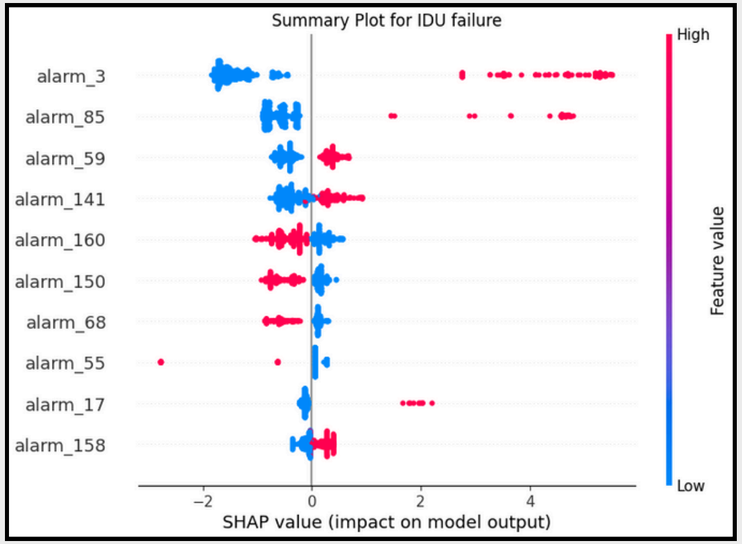
SHAP (SHapley Additive exPlanations) [1] is a unified framework for interpreting machine learning model predictions by assigning each feature an importance value, based on cooperative game theory

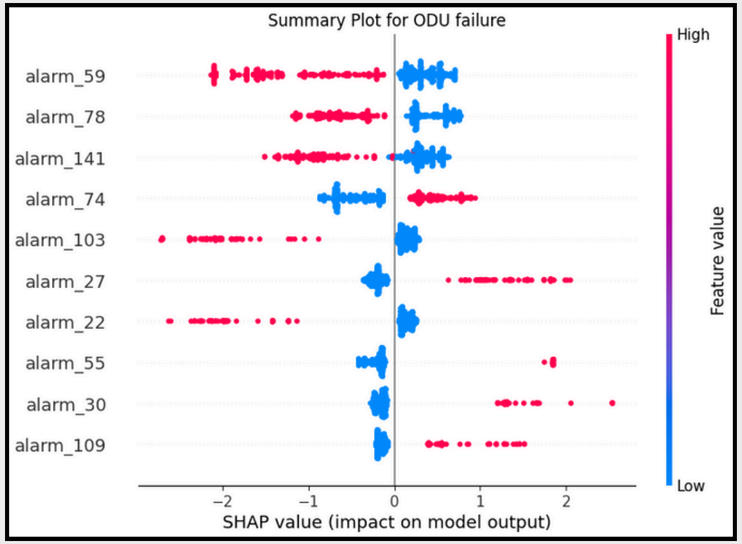
Permutation Feature Importance [2] evaluates the importance of a feature by measuring the change in model performance when the feature's values are randomly shuffled, indicating its impact on predictions.

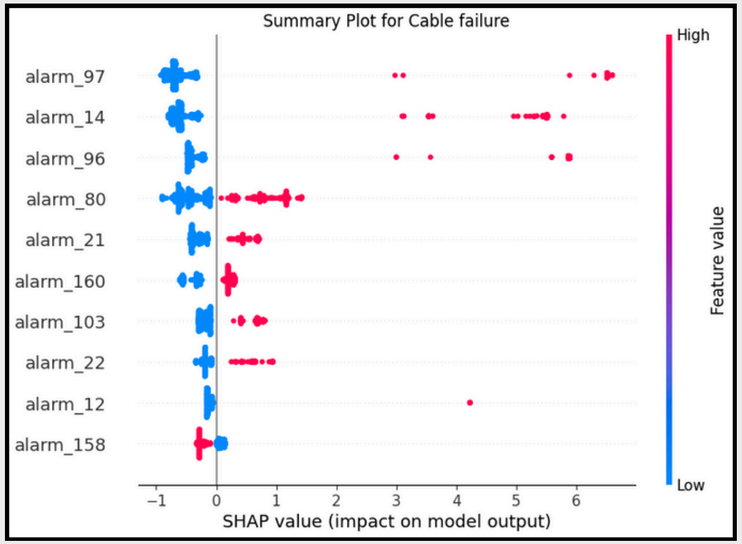
[1] https://shap.readthedocs.io/en/latest/
[2] https://scikit-learn.org/stable/modules/permutation_importance.html

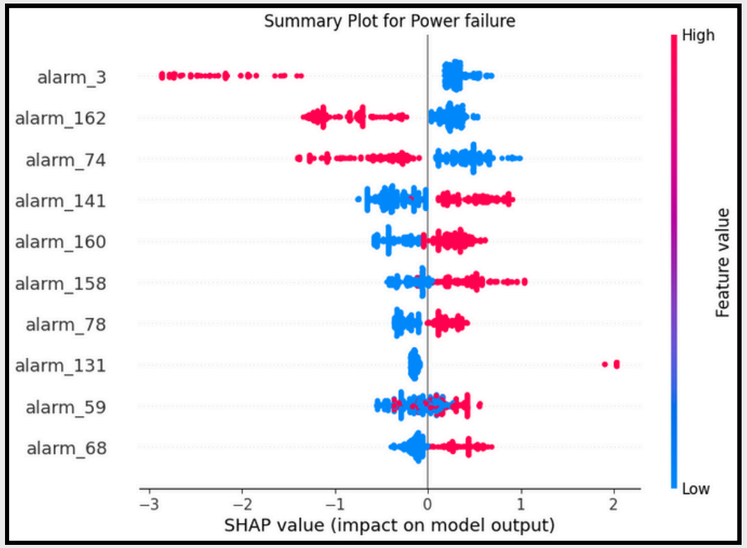
Shap


Summary Plot
alarm_3
alarm_59
alarm_141
alarm_97
alarm_160
alarm_14
alarm_85
alarm_74
alarm_78
alarm_80
alarm_21
alarm_103
alarm_96
alarm_162
alarm_158
alarm_22
alarm_27
alarm_150
alarm_68
alarm_148
Class 1
Class 0
Class 2
Class 3
0.0
0.5
1.0
1.5
2.0
2.5
3.0
mean(|SHAP value|) (average impact on model output magnitude)

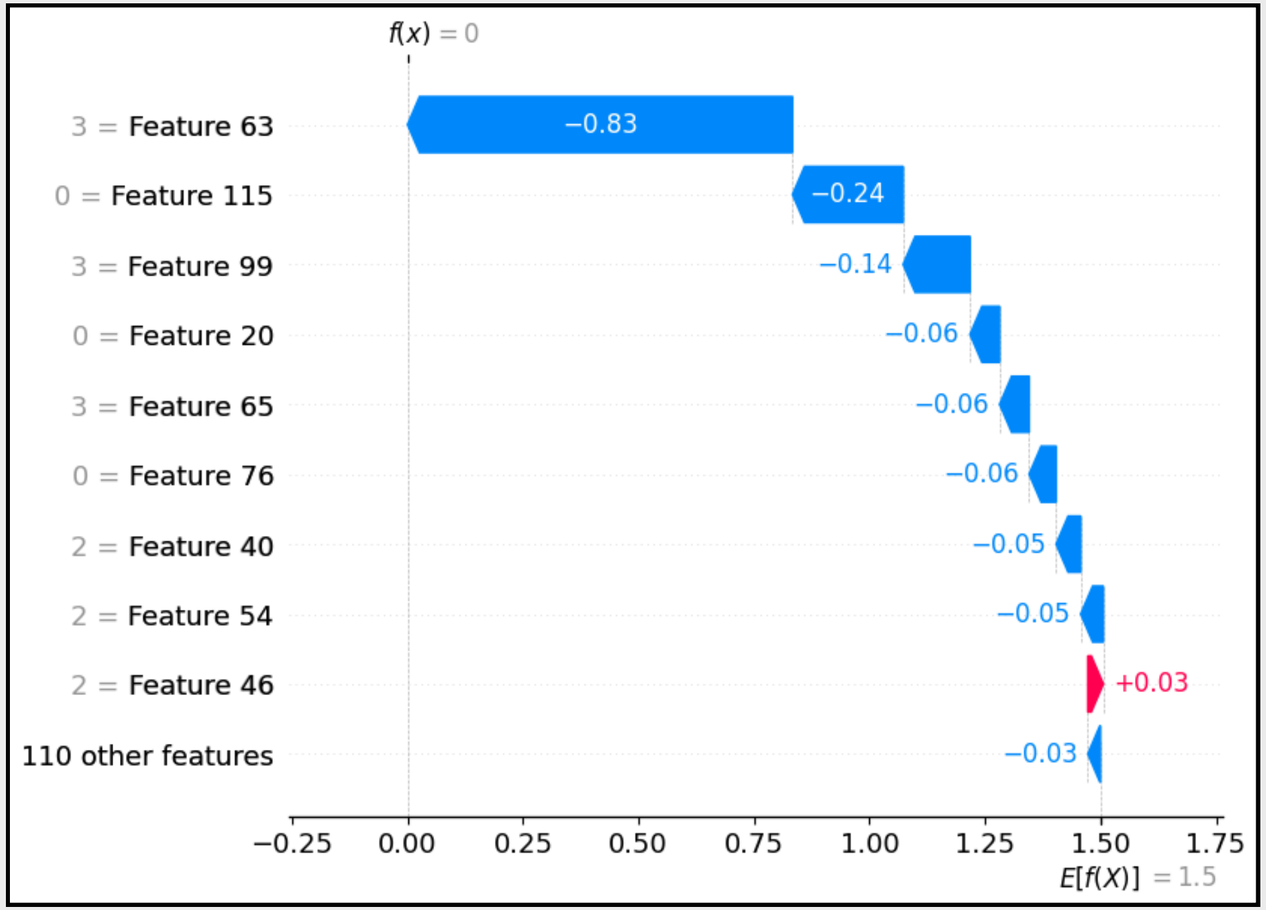

Summary Plot for IDU failure
alarm_3
alarm_85
alarm_59
alarm_141
alarm_160
alarm_150
alarm_68
alarm_55
alarm_17
alarm_158
High
Feature value
Low
−2
0
2
4
SHAP value (impact on model output)

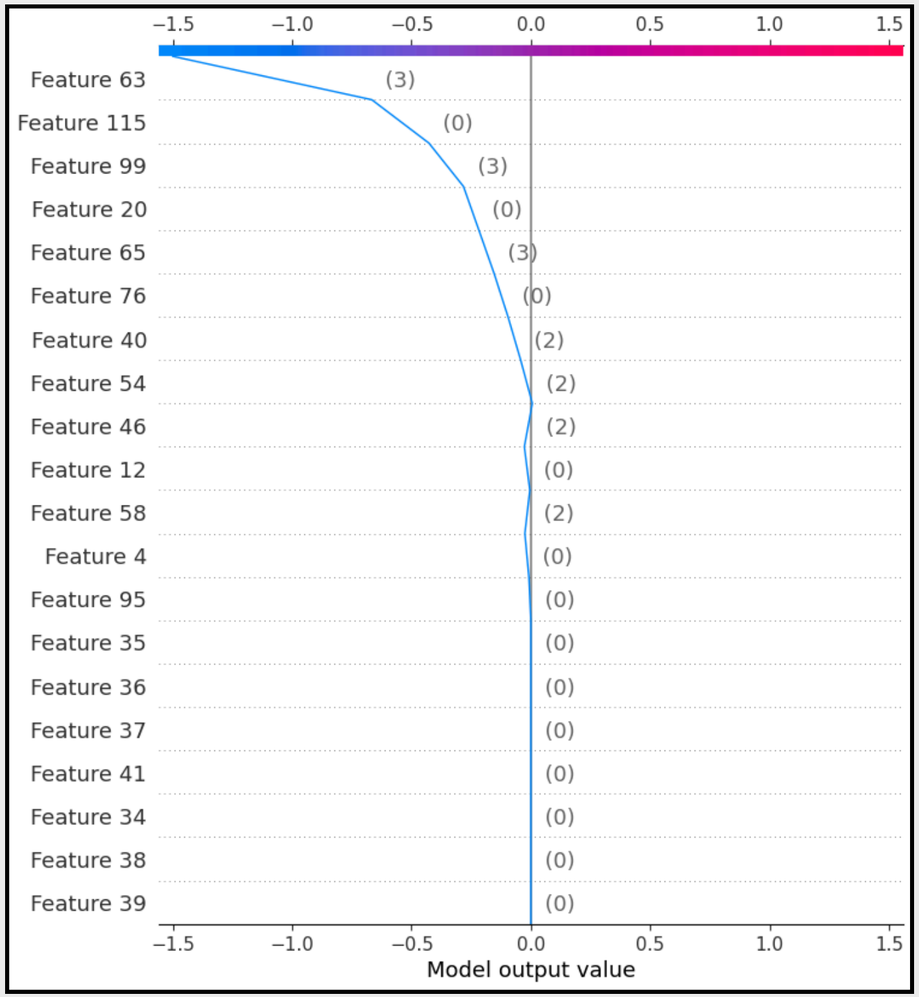

Summary Plot for ODU failure
alarm_59
alarm_78
alarm_141
alarm_74
alarm_103
alarm_27
alarm_22
alarm_55
alarm_30
alarm_109
High
Feature value
Low
−2
−1
0
1
2
SHAP value (impact on model output)


Summary Plot for Cable failure
alarm_97
alarm_14
alarm_96
alarm_80
alarm_21
alarm_160
alarm_103
alarm_22
alarm_12
alarm_158
High
Feature value
Low
−1
0
1
2
3
4
5
6
SHAP value (impact on model output)


Summary Plot for Power failure
alarm_3
alarm_162
alarm_74
alarm_141
alarm_160
alarm_158
alarm_78
alarm_131
alarm_59
alarm_68
High
Feature value
Low
−3
−2
−1
0
1
2
SHAP value (impact on model output)

Shap





Sample n. 100 (Test Set): Predicted Class = 0, True Class = 0

Permutation Feature Importance