Analysis of AUC scores for one-vs-all classifiers in Design report of the in-flight load monitoring model

Summary

In-flight load monitoring model was designed to predict the flight maneuvers which are categorized as following: Ascending, Descending, Left turn, Right turn. To classify each maneuver in the multi-class dataset, one-vs-all classifiers were applied. One-vs-all classifiers are trained to classify a single class from other classes. Instead of letting a single classifier classify multiple classes, an ensemble of one-vs-all classifiers enables paying extra attention to features of each class. Each one-vs-all classifier was validated by AUC score.

	Ascending	Descending	Left turn	Right turn
Original	0.9807	**0.8362**	0.9478	0.9971
PCA	0.95	**0.8477**	0.9594	0.9957

Problem Statement

AUC (Area Under Curve) is relatively low for a one-vs-all classifier that discriminates “descending” class from the other classes.

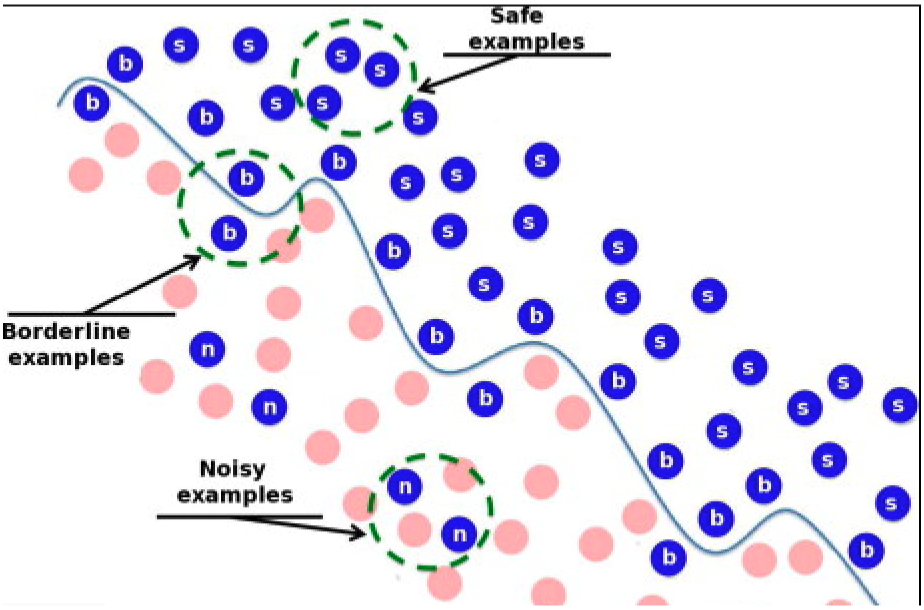
Explanation

AUC is the area under the graph between false positive rate and true positive rate. AUC is maximized if the ratio between false positive rate and true positive rate is the highest. Therefore, AUC will be high when false positive rate is low when true positive rate is high, implying high AUC score indicates that the model performs quite well on classifying positive cases and negative classes. Thus, the model that classifies “descending” from other actions needs improvement. Since the biggest issue in the dataset was explicitly specified to be imbalanced classes, it can be assumed that the relatively low or high number of data labeled as “descending” is higher than data labeled as other classes, thereby it’s necessary to deploy more effective methods to address imbalanced dataset issue than SMOTE.

Solution

Borderline-SMOTE would be the most appropriate choice to strengthen the boundary between the “descending” class (positive) and the other classes (negative). SMOTE simply generate synthetic samples based on any subset in an undersampled class even when the subset are noisy that it rather disturbs the model discriminating positive class from negative class, which leads to higher computational cost.

Borderline-SMOTE, on the other hand, is computationally efficient because it only generates the synthetic samples on the “DANGER” set. “DANGER” set refers to the border between the positive and negative class and are exemplified as “Borderline examples” in Figure 1.



The reason why AUC score is low might be that there are neighborhoods in data distribution that are mixed up by positive and negative classes and are nearly indistinguishable. Borderline-SMOTE, by focusing on generating new samples on borders, simplifies the classification in indistinguishable subset.