Predicting the Incidence of Cheating in FPS Games

Robin Stopa

OVERVIEW

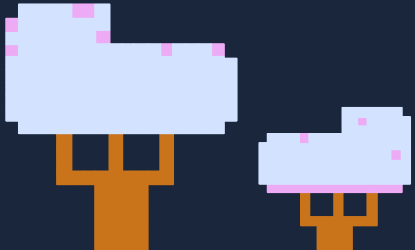
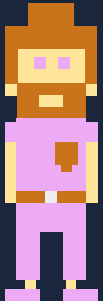
Cheating/ hacking in video games is a large problem, especially in First-Person Shooter games

People use external cheat software such as aimbot and wall-hacks

Game companies want to create an environment where players cannot use cheats to gain a competitive advantage. They spend a lot of resources on anti-cheat software, which identifies cheaters and penalizes them

NEW QUEST

Your task is to create a model which predicts the incidence of cheating in FPS games given characteristics of the game

DATA GATHERING

Unknowncheats

“UnKnoWnCheaTs is the oldest game cheating forum in existence, leading the game cheating community for over 20 years”

Wikipedia

Game description

Levvvel

“Ensuring no one can unfairly gain an advantage over their opponent and more...”

UC post count — **Wiki content** — **Anti-cheat software**

Pub Dev Same

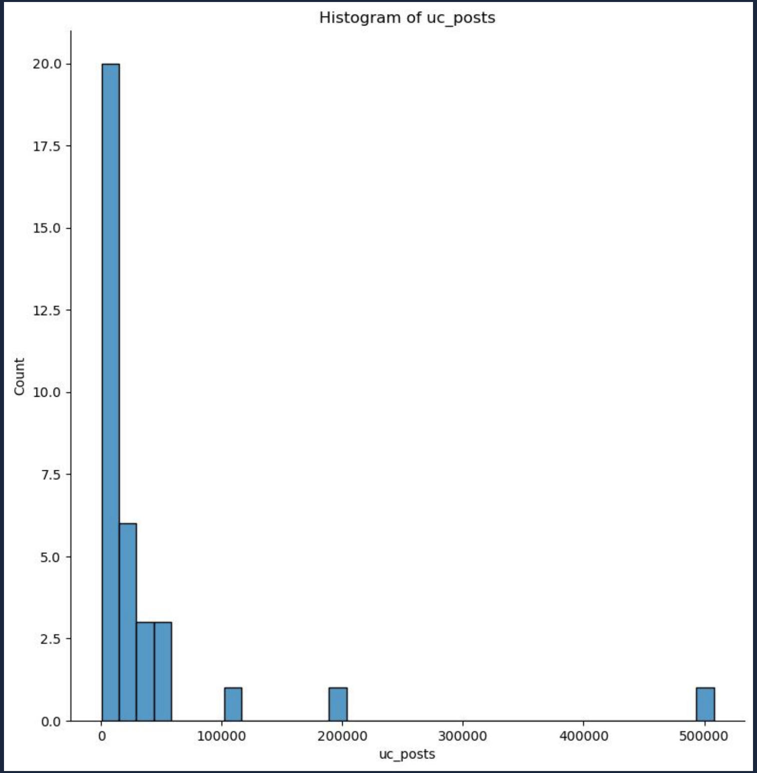
Limitations of Small Data Size

- Final data contained 35 FPS game series
- … there are only so many FPS games where enough people play to warrant having a cheating forum
- Can we gain any quantitative insight via modeling, despite the small data size?

Methods for Working with Small Datasets

1. Simple model: Linear Regression
2. Careful selection of random seed in train-test split
3. SMOGN

INTERESTING RARE DATA



- *Counter Strike* had 500k posts on unknowncheats
- Average posts in dataset was ~46k
- *Battlefield* Series, *PUBG: Battlegrounds*

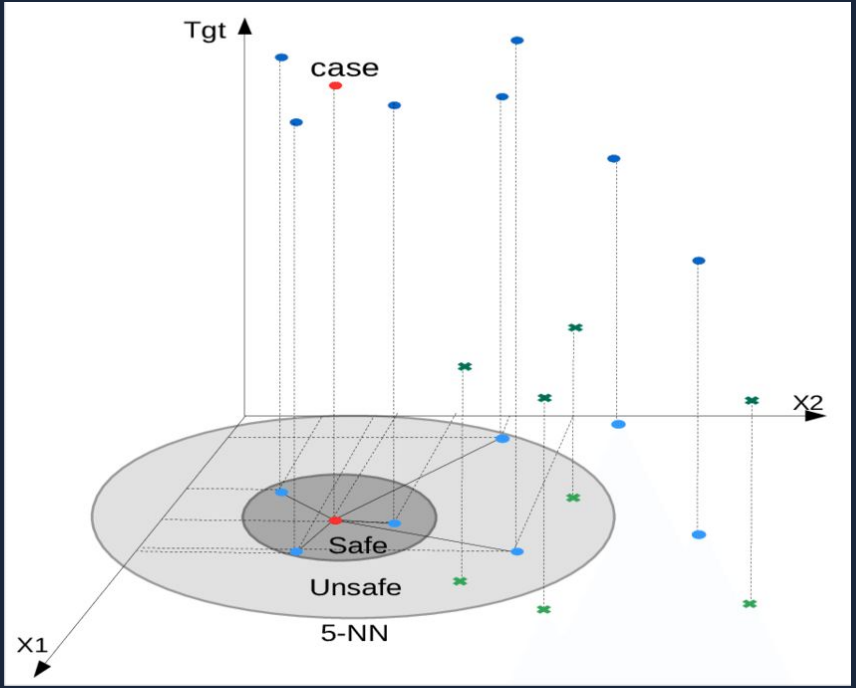
- Want to keep these rare cases because we hope to identify what gaming characteristics lead to having extreme incidences of cheating

SMOGN

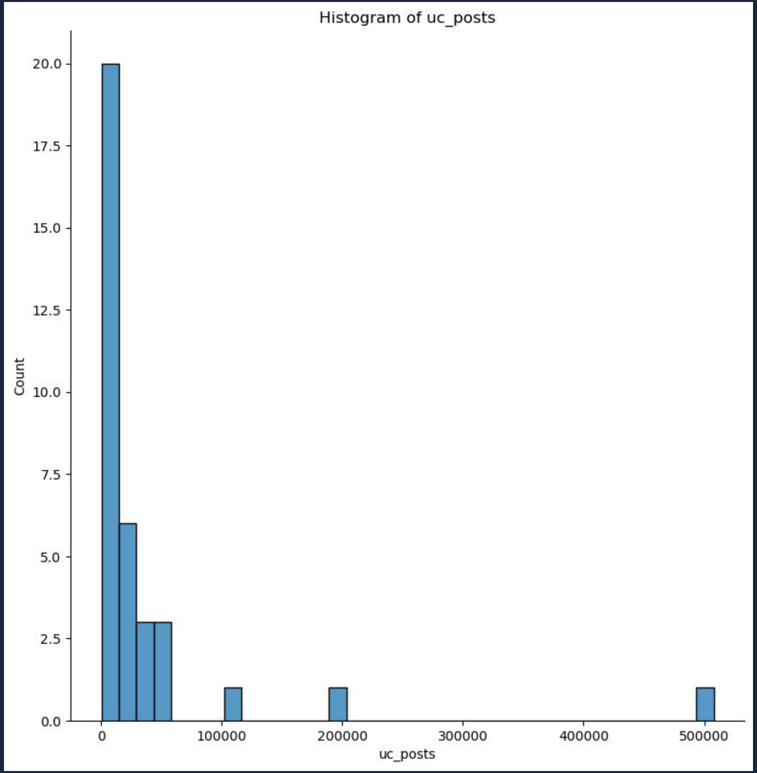
Synthetic Minority Over-Sampling Technique for Regression with Gaussian Noise

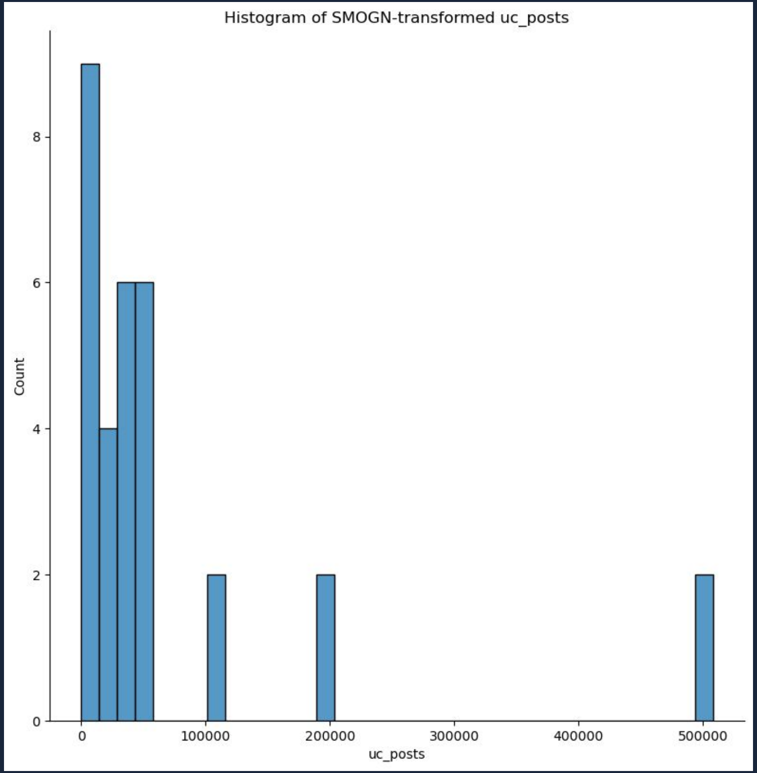
- Pre-processing method for dealing with small data with imbalanced domains
- Splits y into two categories, “Normal” and “Random”
- Samples from data based on category
    - “Normal” cases: random under-sampling
    - “Rare” cases: uses two methods for over-sampling:
        1. Defines a rare case as a seed
        2. Randomly selects K-nearest neighbor
        3. If this neighbor is close, creates new data as weighted average of seed and neighbor (SMOTER)
        4. If this neighbor is not close, creates new data by adding Gaussian noise to seed

“Synthetic example of SMOGN algorithm”



Original target vs. SMOGN-applied target





COMPARING LINREG MODELS

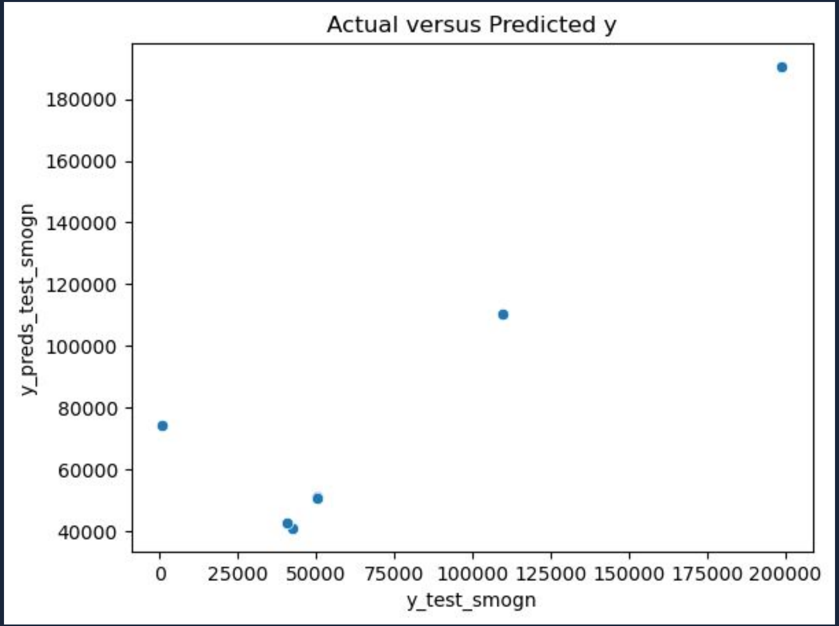
ORIGINAL DATA

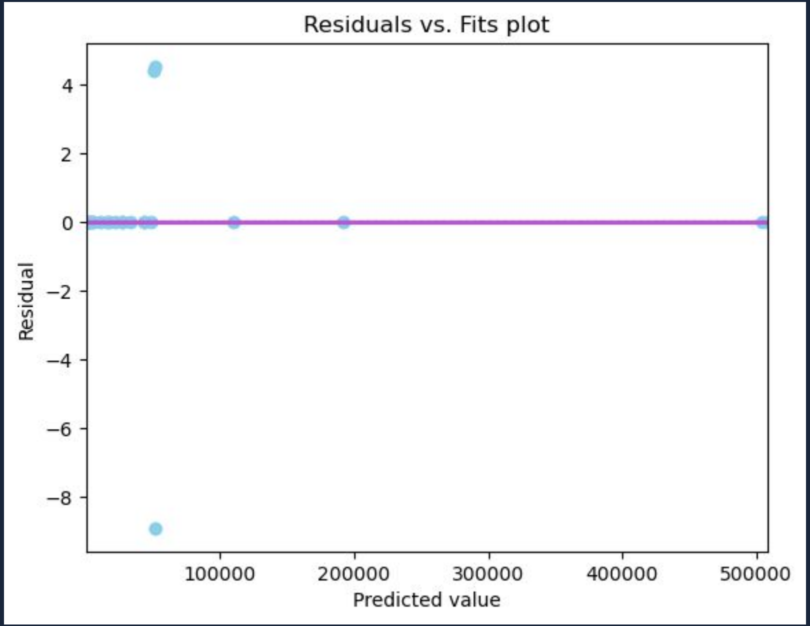
- Average y: ~ 37,000
- Train score: 1.0
- Test score: -0.348

SMOGN DATA

- Average y: ~ 71,000
- Train score: 1.0
- Test score: 0.785

Actual vs. Predicted (SMOGN-applied model)





RMSE for test: 27890
RMSE for train: 2.2

MODEL RESULTS

- "Counterstrike" and "Valve anti-cheat" have the highest coefficients and indicate high incidence of cheating (*Counterstrike* has the highest unknowncheat post count, and it uses Valve anti-cheat)

- Vanguard anti-cheat has a very high negative coefficient and therefore is indicative of low incidence of cheating

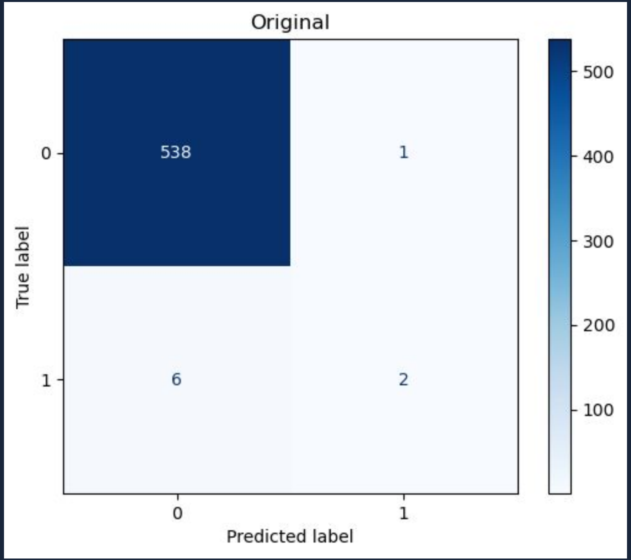
- Punkbuster and Easy anti-cheat: negative coefficients

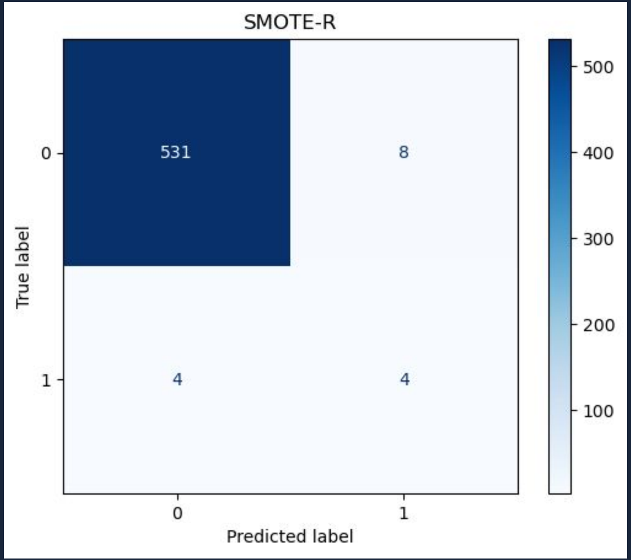
- If the publisher and developer are the same: indicative of a low incidence of cheating

Our model is still very overfit so these cannot be offered as concrete solutions

SMOTE ON ANOTHER DATASET

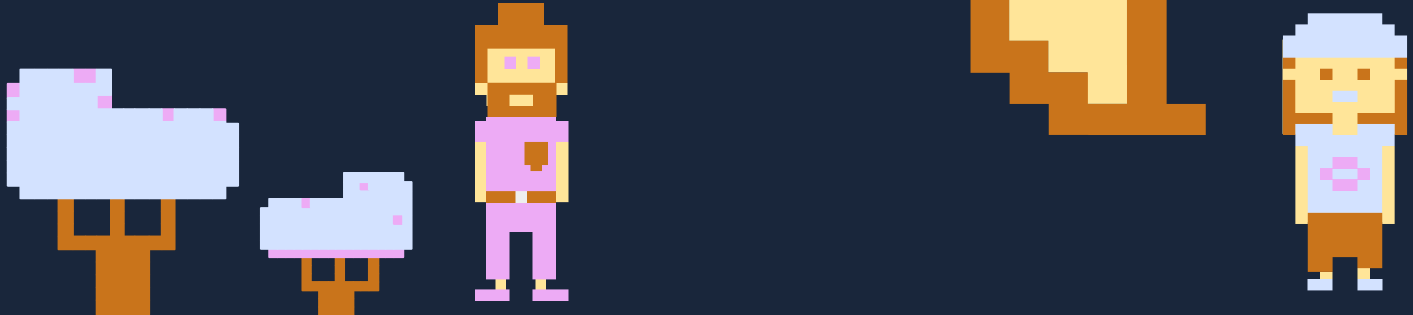
- Video game sales data (n = 1,907)
- Contains: platform, genre, publisher, review, sales in countries, global sales
- Turned into binary classification problem (global sales = 10M+, 2% incid.)
- Compared logreg on original data vs. SMOTE-R preprocessed data





ADDRESSING THE PROBLEM STATEMENT

- Find more data - embellish the dataset with other sources and keep using SMOGN to learn about cases with high incidences of cheating
- Break apart game series into individual games, while being able to account for the similarity their wikipedia entries may have
- Ask companies to share more about their anti-cheat software to collaboratively work towards a better gaming community



CONCLUSION

Our real problem statement was really:

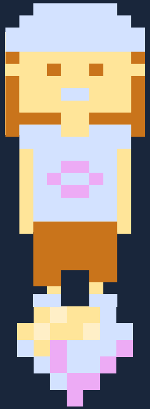
Can we use pre-processing techniques to create viable models from from small, imbalanced data?

Unknowncheats data

⇛ **SMOGN is a valuable pre-processing tool for imbalanced regression**

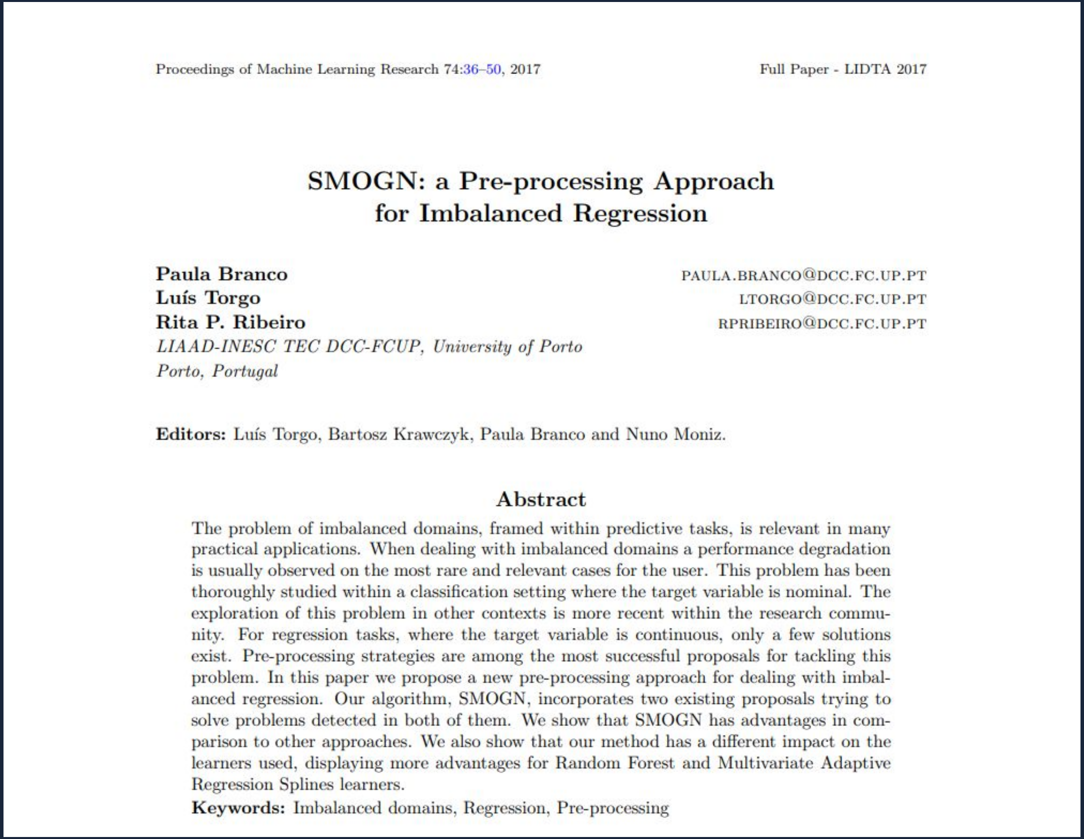
Video Game Sales data

⇛ **SMOTE-R is a valuable pre-processing tool for imbalanced classification**



APPENDIX

SMOGN RESOURCE



Proceedings of Machine Learning Research 74:36–50, 2017 Full Paper - LIDTA 2017

SMOGN: a Pre-processing Approach for Imbalanced Regression

Paula Branco PAULA.BRANCO@DCC.FC.UP.PT
Luís Torgo LTORGO@DCC.FC.UP.PT
Rita P. Ribeiro RPRIBEIRO@DCC.FC.UP.PT
LIAAD-INESC TEC DCC-FCUP, University of Porto
Porto, Portugal

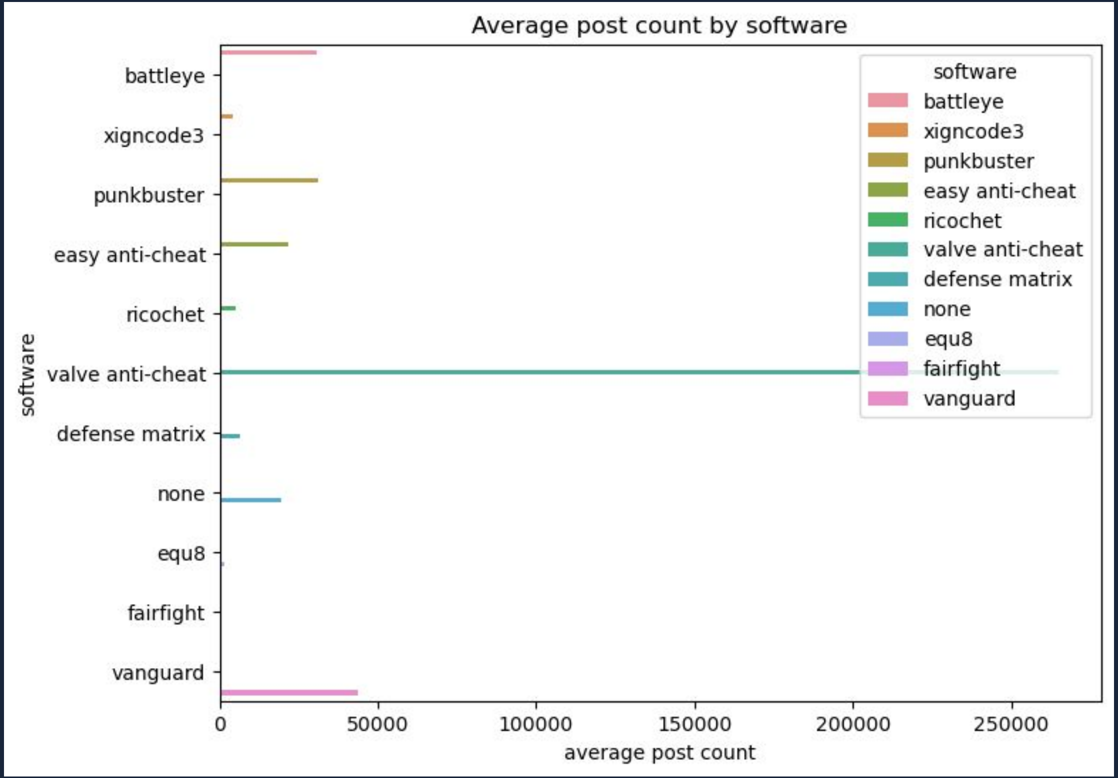
Editors: Luís Torgo, Bartosz Krawczyk, Paula Branco and Nuno Moniz.

Abstract

The problem of imbalanced domains, framed within predictive tasks, is relevant in many practical applications. When dealing with imbalanced domains a performance degradation is usually observed on the most rare and relevant cases for the user. This problem has been thoroughly studied within a classification setting where the target variable is nominal. The exploration of this problem in other contexts is more recent within the research community. For regression tasks, where the target variable is continuous, only a few solutions exist. Pre-processing strategies are among the most successful proposals for tackling this problem. In this paper we propose a new pre-processing approach for dealing with imbalanced regression. Our algorithm, SMOGN, incorporates two existing proposals trying to solve problems detected in both of them. We show that SMOGN has advantages in comparison to other approaches. We also show that our method has a different impact on the learners used, displaying more advantages for Random Forest and Multivariate Adaptive Regression Splines learners.

Keywords: Imbalanced domains, Regression, Pre-processing

Average Post Count by Anti-Cheat Software


Average post count by software
software
battleye
xigncode3
punkbuster
easy anti-cheat
ricochet
valve anti-cheat
defense matrix
none
equ8
fairfight
vanguard
average post count