Automated Outlier Treatment for Improving Predictions

Akiva Kleinerman

March 2023

1 Abstract

Outliers are outlying observations: values of features that significantly differ from other features of samples in a dataset. Outliers often have a negative effect on the prediction accuracy of machine learning models, and therefor it is common to apply a process of outlier treatment before training model. However, there is no "one fits all" solution, and to date, to the best of our knowledge, there is no automated process that compares the method or combines various methods. This often makes the treatment of outliers a time-consuming "trial and error" process and in many cases a non-optimal method is applied. In this work we propose a novel method that combines univariate outliers detection and treatment methods and integrate it in a new system that compares many outliers handling methods. We found that the novel automated method for univariate outliers outperforms all multivariate methods and the baseline method.

2 Problem Description and Background

There are a few different methods for detecting and treating outliers. Outlier *detection* methods can be divided into two different main approaches: 1) Multivariate outlier detection - A multivariate outlier is an outlier in a combination of dimensions (i.e. features); and 2) Univariate outlier detection- in this approach, each dimension (i.e. feature) is examined separately, and the outliers are the samples that have outlying values for the examined dimension. [3, 12]. Within the univariate outlier detection approach, previous work make a distinction between *single-step* and *sequential* procedures for outlier detection. Single-step procedures identify all outliers at once as opposed to successive elimination data. In the sequential procedures, at each step, one observation is tested for being an outlier.

After the detecting of outliers in a dataset, some sort of *treatment* for the outlier in order to try improve the prediction accuracy. A common method is to simply drop the samples that were identified as outliers. However, this method can lead to lose of important data [3]. Additional common methods or *replacing* the outlying values with the average value in the dimension or some other value (capping and flooring) [3]. Another option is transforming *all* values of dimensions with outliers, such as log transformation [4] .

However, there is no optional "one fits all" solution [5,6,11]. The outlier treatment depends on the cause of the outlier and in many cases the cause is not clear, even after investigation and visualization. This often makes the treatment of outliers a time-consuming "trial and error" process, where the data scientist iteratively tries various methods and evaluates them.

In this work we present a system that automates the process by searching for the best combination of detection and treating methods in terms of accuracy of a prediction model. We present the performance of various methods on four tabular data-sets and compare the their performance.

3 The Solution

The system performs two separate processes that correspond to the two outlier handing approaches: 1) Multivariate outlier handling process and 2) Univariate outlier handling process. In each process we combine various detection and treatment methods and finds the best combination in terms of accuracy. Below we describe each process separately. In this work, we use the XGBoost algorithm [16] for the prediction model, since this algorithm has recently shown superior performance [16]. for the sake of simplicity- we assume the primary accuracy measures are area under curve (AUC) for classification problem and average absolute distance (AAD) for regression problems. The model and accuracy measure can be easily configured according to the users' needs and concerns.

3.1 Multivariate Outlier Handling

In this process, the system applies five common multivariate outlier detection methods:

1. Isolation forest - isolates' observations by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of the selected feature [9].

2. Ellipicit envelope: Elliptic Envelope is intuitively built on the premise that data comes from a known distribution. If we draw an ellipse around the gaussian distribution of data, anything that lies outside the ellipse will be considered an outlier [2]

3. local outlier factor- an unsupervised anomaly detection method which computes the local density deviation of a given data point with respect to its neighbors [1].

4. One class SVM: an unsupervised algorithm that learns a decision function for novelty detection: classifying new data as similar or different to the training set [10].

5. Gaussian mixture models: A Gaussian Mixture Model (GMM) is a probabilistic clustering model that assumes each data point belongs to a Gaussian distribution [13]. A GMM can be used detect outliers by identifying the data points in low-density regions [8].

All of these methods are implemented in the sklearn library and imported in the project. The process iterates over all methods- for each method it detects the outlier samples. Then, all of outliers are removed from the dataset. Note that in this process the only treatment that is considered is removing the outlier. Then the process trains and evaluates the accuracy of the model on the filtered data . Then, the results are sorted according to the accuracy and find the best method and its accuracy.

3.2 Univariate outlier detection

For the univariate outlier detection approach, we apply the following methods:

1. interquartile range (IQR) anomaly detection - in this method the values of a feauture are divided into four corresponding intervals based upon the. A quartile is what divides the data into three points, Q1,Q2 and Q3, and four intervals. IQR is the difference between the third quartile and the first quartile (IQR = Q3 -Q1). Outliers in this case are defined as the observations that are below $Q1 - 1.5 \times IQR$ or above $Q3 + 1.5 \times IQR$ [15].

2. Median absolute deviation (MAD) based outlier. MAD is a measure of the variability of a distribution, similar but different than mean standard deviation. In this outlier detection method- the samples that are further than a given threshold of absolute deviations from the median are considered outliers. The exact threshold can be configured. This method has been found to be superior to the correspanding method that uses median standard deviation [7].

3. Percentile based outlier - this method simply divides the data into percentiles and defines all data above a certain percentile and below a certain percentile as outliers.

We chose these detection methods because they are commonly used. All the methods include thresholds for identifying outliers. In this work we used constant threshold values: more information in supplementary material.

3.3 Outlier Treatment methods

In this work we try three common methods for treating univariate outliers :

- —Dropping outliers- meaning that the whole sample is erased for the training set.
- Quantile based *flooring and capping*; in this technique,we can do the flooring for the lower values and capping for the higher values
- *imputation*: replacing the outlier values by the median value of the columns. We use Median and not the mean since the mean is highly influenced by the outlier values.

3.4 Iterative Univariate Outlier Handling (IUOH)

The iterative process is described in Algorithm 1 below. The process performs iterations of outlier handling, in each iteration (starting in row 4 in the algorithm) one column is treated. In the iteration- the process first finds the feature (within the features that have not been treated yet) with the highest "outlier treatment benefit", meaning the feature that treating the outlier will give the highest improvement in accuracy. To achieve this the process (rows 10- 14) trains and evaluates the model after every possible combination of outlier detection method and treatment method. If the highest accuracy in the current iteration is better than the accuracy in the precious iteration: the process takes the feature that received the highest accuracy, and detects and treats the feature according to the best detection and treatment combination in terms of accuracy (rows 18-22). (Recall that we assume there is a single measure of accuracy that is the primary focus of the user.) The process ends when the current iteration did not find any improvement by treatment of the outlier.

Algorithm 1 Iterative Univariate Outlier Handling (IUOH)

Data: a dataset: D , a prediction model : P
Result: the accuracy of the model on the dataset after the process

```
1  best-accuracy-global ← 0
2  remaining-features← D.features
3  improved ← True
4  while improved do
5      improved ← False
6      D-temp ← D.copy()
7      best-accuracy-iteration ← 0
8      forall feature in remaining-features do
9          accuracies-list ← {}
10         forall (detection, treatment) in (Detection-methods × Treatment-methods) do
11             D-temp← treat-outliers(treatment, D − temp, feature, outliers)
12             accuracy← fit-and-evaluate(P, D-temp)
13             accuracies-list.append( [accuracy, detection, treatment])
14         best-accuracy, detection, treatment= get-best(accuracies-after-treatments)
15         if best-accuracy > best-accuracy-iteration then
16             best-accuracy-iteration← best-accuracy
17             best-feature, best-detection, best-treatment ← feature, detection, treatment
18     if best-accuracy-iteration > best-accuracy-global then
19         best-accuracy-global← best-accuracy-iteration
20         improved ← True
21         remaining-features.remove(best-feature)
22         D ← treat-outliers(treatment, detection, D, best-feature)
23 return best-accuracy-global
```

4 Experimental Evaluation

4.1 The Datasets

For our evaluation we used four datasets: two of a classification model and two of a regression model. The data set are all public, we will describe them shortly:

1. **NBA rookie** : the dataset includes NBA rookie statistics and a label indicating if the rookie will last 5 years in the NBA league (binary classification). Includes 1340 samples and 19 features. The dataset and further details can be found in: *https://www.kaggle.com/competitions/iust-nba-rookies/overview*.

2. **Breast Cancer**: The dataset includes features that are computed from a digitized image of a fine needle aspirate of a breast mass and a label indicating if the cancer is benign or malignant (binary classification). They describe characteristics of the cell nuclei present in the image. Includes 570 samples and 30 features. The dataset and further details can be found in: *https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data*.

3. **US college graduates**- the dataset includes a 18 features description of 777 colleges in the USA. The label is the graduation rate in percentage. Since the value of the label are continuous, we will use regression models. The dataset and further details can be found in: *https://www.kaggle.com/yashgpt/us-college-data*.

4. **Life expectancy**: The dataset includes features describing the health status as well as many other related factors in countries during a specific year. The label is the life expectancy in the country during the year. Since the value of the label are continuous, we will use regression models. Includes 2938 samples and 20 features. The dataset and further details can be found in: *https://www.kaggle.com/datasets/kumarajarshi/life-expectancy-who*.

4.2 The Evaluation Process

The system outputs the detection and treatment methods that were gave the best performance. In addition, we present the size of the outliers in according to each detection method (and visualize the outliers). We will present the performance of our system and the baseline approaches for all datasets. Then we will present the IOUT process in details for two datasets.

For each of the data-sets, we evaluated the performance of our proposed methods, IUOT and MOD, and the performance of the model on the original dataset without outlier treatment. In all data-sets, we used the k-fold cross validation technique [14], in order to promise consistency.

4.3 Results

In the following subsections we will first demonstrate the results of the various method in terms of AUC. Later, we demostrase the IUOT process in detail on one of the datsets.

4.3.1 Methods' Comparison

The results show that In all of the data-sets, *our novel method outperformed the baseline and all multivariate methods.*

dataset	baseline accuracy	multivariate accuracy	multivariate method	univariate accuracy
nba rookie	0.669	0.653	ellipicit envelope	**0.683**
breast cancer	0.951	0.967	guassian model	**0.976**

Table 1: **Comparison of performance of the *classification* prediction model for the various outlier handling methods.** The accuracy measure here is the **area under the curve (AUC)**, and therefor *the higher the better.*

dataset	baseline accuracy	multivariate accuracy	multivariate top method	univariate accuracy
college graduates	11.107	10.09	ellipicit envelope	**9.424**
life expectancy	1.281	1.264	ellipicit envelope	**1.206**

Table 2: **Comparison of performance performance of the *regression* prediction model for the various outlier handling methods.** The accuracy measure here is the **average absolute distance (AAD)**, and therefor *the lower the better.*

4.3.2 IOUT process example

In this subsection we illustrate an example of the process of the IOUT for NBA rookie dataset. Recall that we used k-fold for the evaluation, here we show the process for a randomly chosen fold. For this dataset, the AUC performance of the model *without* handling outliers was 0.669. In the first iteration of the IOUT process (rows 4-22 in Algorithm 1) the process finds that the feature STL, which describes the average number of steals of the rookie, has the biggest outlier handling benefit. Specifically, using the *mda* method, 49 outliers were found. See Figures 1 and 2 for the visualisation of the feature's distribution. The process found that by *dropping* these outlier, the most significant improvement will be achieved, an AUC of 0.673. In the second iteration, the process found that the TOV: the rookie's average of turnovers, has the biggest outlier handling benefit. For TOV, the process found that using the IQR detection method together with flooring and capping for treatment, gave the biggest improvement: a 0.676 AUC, meaning a 0.003 improvement. In the next round, the process finds that detecting outliers in

the AST feature with the *MAD* method and treating with *median replacement* can give the biggest improvement - a 0.009 improvement to a 0.686 AUC. In the next round- the process did not find any improvement within the remaining features, and therefor the process halts. Thus, the final performance is 0.686.

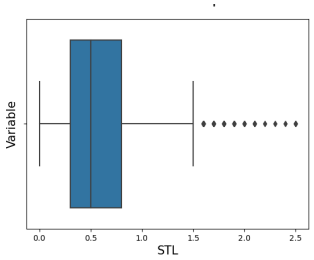

Figure 1: STL boxplot

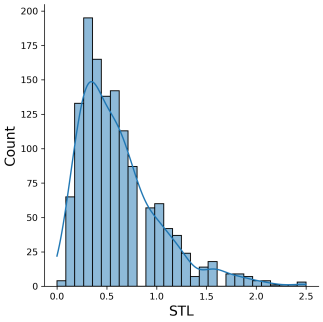

Figure 2: STL Histogram

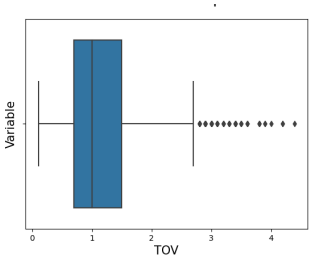

Figure 3: TOV boxplot

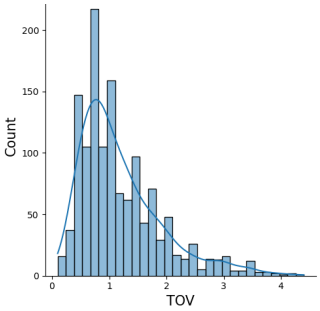

Figure 4: TOV Histogram

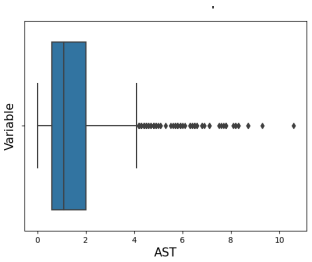

Figure 5: AST Boxplot

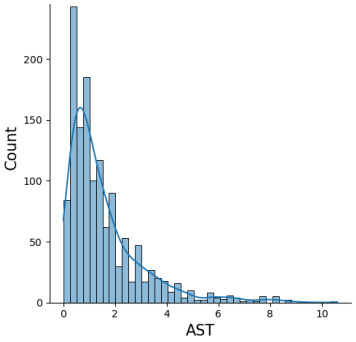

Figure 6: AST histogram

5 Conclusion and future work

In this work we presented and evaluated an automated system for detecting and treating univariate and multivariate outliers with an objective of improving the accuracy of a prediction model. Specifically we presented IOUT- a process that evaluates various combinations of univariate detection and treatment methods for outliers and presents the combinations that will give the best performance. We found that IOUT favorably compares to baseline methods and multivariate outlier methods.

It is important to mention the limitations of our work. Our system creates and evaluates multiple prediction models, this process can have a long runtime, especially in large dataset with many features. In order to reduce the runtime, we should consider various approaches for reducing the number of combinations of detection and treatment methods, possibly be heuristics. In addition, in this work we assumed the user gives the prediction model as input to the system. However, it is possible that a different model can outperform the given model after treating the

outliers. Therefore, in future work we would like to integrate evaluation of several models with the outlier treatments, so that the system could also recommend on a model that will best fit. In addition, in this work we gave constant values to the various threshold in the outlier detection method. However, the methods can be very sensitive to the threshold values. Therefore, we would include thresholds exploration in the next version as well.

References

[1] Omar Alghushairy, Raed Alsini, Terence Soule, and Xiaogang Ma. A review of local outlier factor algorithms for outlier detection in big data streams. *Big Data and Cognitive Computing*, 5(1):1, 2020.

[2] Mohammad Ashrafuzzaman, Saikat Das, Ananth A Jillepalli, Yacine Chakhchoukh, and Frederick T Sheldon. Elliptic envelope based detection of stealthy false data injection attacks in smart grid control systems. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1131–1137. IEEE, 2020.

[3] Irad Ben-Gal. Outlier detection. *Data mining and knowledge discovery handbook*, pages 117–130, 2010.

[4] Denis Cousineau and Sylvain Chartier. Outliers detection and treatment: a review. *International Journal of Psychological Research*, 3(1):58–67, 2010.

[5] Rémi Domingues, Maurizio Filippone, Pietro Michiardi, and Jihane Zouaoui. A comparative evaluation of outlier detection algorithms: Experiments and analyses. *Pattern recognition*, 74:406–421, 2018.

[6] Christophe Leys, Marie Delacre, Youri L Mora, Daniël Lakens, and Christophe Ley. How to classify, detect, and manage univariate and multivariate outliers, with emphasis on pre-registration. *International Review of Social Psychology*, 32(1), 2019.

[7] Christophe Leys, Christophe Ley, Olivier Klein, Philippe Bernard, and Laurent Licata. Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of experimental social psychology*, 49(4):764–766, 2013.

[8] Lishuai Li, R John Hansman, Rafael Palacios, and Roy Welsch. Anomaly detection via a gaussian mixture model for flight operation and safety monitoring. *Transportation Research Part C: Emerging Technologies*, 64:45–57, 2016.

[9] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE, 2008.

[10] Larry M Manevitz and Malik Yousef. One-class svms for document classification. *Journal of machine Learning research*, 2(Dec):139–154, 2001.

[11] Fabrice I Mowbray, Susan M Fox-Wasylyshyn, and Maher M El-Masri. Univariate outliers: a conceptual overview for the nurse researcher. *Canadian Journal of Nursing Research*, 51(1):31–37, 2019.

[12] Daniel Peña and Francisco J Prieto. Multivariate outlier detection and robust covariance matrix estimation. *Technometrics*, 43(3):286–310, 2001.

[13] Douglas A Reynolds et al. Gaussian mixture models. *Encyclopedia of biometrics*, 741(659-663), 2009.

[14] Juan D Rodriguez, Aritz Perez, and Jose A Lozano. Sensitivity analysis of k-fold cross validation in prediction error estimation. *IEEE transactions on pattern analysis and machine intelligence*, 32(3):569–575, 2009.

[15] Neil C Schwertman, Margaret Ann Owens, and Robiah Adnan. A simple more general boxplot method for identifying outliers. *Computational statistics & data analysis*, 47(1):165–174, 2004.

[16] Ravid Shwartz-Ziv and Amitai Armon. Tabular data: Deep learning is not all you need. *Information Fusion*, 81:84–90, 2022.