Identifying Bias in a Multi-Class Model using Confusion Matrices

##### Disha Kalal

Faculty of Science

Queensland University of Technology

2, George Street, Brisbane, 4001, QLD, Australia

###### *Abstract* - Machine learning models are widely used in this world. From grocery shopping to which movie to watch, algorithms impact both our decisions and our well-being. However, these systems are not always incorrect. Moreover, machine learning models tend to learn bias which is present in historical data which can lead to unfair outcomes. If we want to build models more responsibly, then we need to build tools/libraries that help in the in-depth validation of these models. This article uses an open-source python confusion matrix (PYCM) library to evaluate the performance of a classifier and detect bias in a classifier. PYCM library offers overall statistics, individual class statistics, and visualization of the confusion matrix. The implemented set of functions and statistics enables us to detect bias in the classification model and validate its performance. The article will emphasize multi-class classification model performance and provide more information on class statistics and overall statistics of the classification model.

###### *Keywords— PYCM, Multi-class, classification, Bias, machine learning*

1. INTRODUCTION

Machine learning and in particular evaluation of artificial intelligence models are gaining attention within the data science community. The reason for this is that machine learning or predictive models are becoming more and more in use in our lives. The use of predictive models is more significant in areas such as online shopping, movie recommendations, loan decisions, facial recognition to unlock phones, etc. Some systems make use of automated techniques that learn historical data or biased data. Whether giving loans or seeking job offers, attributes such as age, sex, ethnicity, etc. might impact the decision. As we all can expect this can lead to unfair outcomes for less privileged groups or fair outcomes for more privileged groups vice versa. Bias is an unreasonable weight or decision in favor of or against an idea, usually in a way that is unfair [1].

When there are poor strategies, historical reasons for this to happen, such outcomes are unacceptable for legit reasons.

Bias in machine learning models has potentially many different origins. In [2] authors discussed 5 types of bias which are *algorithmic* *bias is the error that occurs when an algorithm produces results that are unfair due to erroneous assumptions in the machine learning process. sample bias, prejudice bias, and measurement bias*. This shows how many different bias origins exist which are potentially hidden in data itself. Whether one would like to address it or not, it is important to detect bias in a machine learning model and make fair decisions. Sometimes using a system that has a bias in it can result in greater harm. This was

addressed by Ethe Europe council that wrote in the set of rules

where it is mentioned that usage of facial recognition for the sake of determining age sex, origin, etc., should be prohibited [3].

Measuring the performance of a binary classification model is much easier and many different metrics such as recall, F-score, and precision are already in use. This paper is mainly focusing on detecting bias and evaluating the performance of a multi-class classification model using confusion matrices.

The confusion matrix is a matrix that plots the amount of correct prediction against the amount of incorrect prediction [6].

A picture containing timeline

Description automatically generated

Figure 1: Confusion matrix

The dataset that is used in this project is downloaded from open ml [4]. I have used the medical dataset because it is important to identify bias in such models cause the impact of an incorrect decision can lead to the wrong diagnosis. A lymph dataset has three unique classes that are fibrosis means thickening or scarring of tissues, malign\_lymph which means it contains cancer cells and metastases means the spread of cancer from the place it started to other parts of the body.

In this paper, I have answered the research question on how to detect bias in a classification model using PYCM? The paper explains in-depth methodology on how algorithmic bias is detected in a classification model which is built using a medical dataset. I have used an open-source python confusion matrix (PYCM) library to evaluate and detect bias in the multi-class model [5].

1. RELATED WORK

I have proposed a way to detect bias and evaluate a multi-class classification model using PYCM in this project. Here, I will briefly review prior work on these topics.

Evaluating the performance of predictive models has been taken into consideration and a paper [7] addresses detection and mitigation of bias in a predictive model. The paper was performed on a binary class classification model. Model validation must be easy to understand. Many frameworks have been introduced in Python to evaluate fairness criteria, few of those are aif360 [8] or aequitas [9]. The above-stated frameworks have various features for detection, mitigation, and visualization of bias in machine learning models. For R language, in paper [10] confusR package has been introduced which proposes a way to visualize the empirical performance of multinomial classification systems using odd ratios to interpret their confusion matrices. ConfusR did an exceptional job in visualizing prior and posterior odds but I kind of lacked understanding of R language hence I opted to look for more papers where bias detection was performed in python language. The package fairmodels [7] proposes a comparison between models and people, also gives direct feedback on if the model is fair or not. The package also has an option of customizing fairness objects according to our needs. This package only focused on binary class models and did not address or give in-depth information on how fairmodels tools perform on a multi-class model. The package has very complex formulas to understand. The framework fairlearn [11] has various features for detection and visualization of bias in a binary-class model but it does not address how the framework performs in case of a multi-class classification model.

In the article [5] python confusion matrix (PYCM) library has been introduced. PYCM is a multi-class confusion matrix that is implemented in python language that supports direct matrix and input data vectors. PYCM is a tool used for post-classification model evaluation which supports many classes. It supports a broad range of matrix which is derived from confusion matrix to evaluate multi-class models. PYCM has a confusion matrix function that takes weights, threshold, classes, direct matrix as an input and provides an output of overall model statistics and individual class statistics. The output can be exported in a CSV file or an HTML file. PYCM also supports comparison between various classification models.

Diagram

Description automatically generated

Figure 2: PYCM confusion matrix block diagram

The above diagram shows in-depth information about the PYCM confusion matrix. The function uses direct matrix, weights, thresholds, input data vectors as an input and generates overall statistics of the classification model and generates individual class statistics. Confusion matrix function accepts the following data types, list of NumPy array, dictionary, int, float, file object, and bool. I will list down a few of the basic class statistics from the confusion matrix function.

* True positive (TP): The true positive values are the ones when the condition is correctly identified.
* True Negative (TN): The true negative values are the ones that do not identify a condition when the condition is not present.
* False Positive (FP): The false-positive values are the ones that detect the condition when it is not present
* False Negative (FN): The false negative values are the ones when it does not identify the condition when it is present.
* True positive rate (TPR): Also called sensitivity, recall, or probability of detection of few fields. The calculates the proportion of positive outcomes that are correctly identified.
* True negative rate (TNR): Also called specificity. It calculates the negative outcomes that are correctly identified.
* False-positive rate (FPR): It calculates the proportion of negatives that still produces positive results.
* False-negative rate (FNR): It calculates the proportion of positives when it still produces negative results.

In this paper, I will make use of the open-source library PYCM and its confusion matrix function that generates overall and individual class statistics which can help detect algorithmic bias in the classification model.

1. METHODOLOGY

The strategy that I used to answer my research question is I started with understating what bias is in machine learning, what types of bias are present in machine models, and its emergence in AI systems. Further, I started looking for what is the impact of having a biased system in use especially in bigger companies and in medical sectors. Almost after 2 weeks of research and reviewing papers that spoke about similar issues and what kind of approach these papers followed. I started building an idea on how to answer my research question. Keeping in mind the time constraint of the research IT unit I started looking for existing tools/ libraries that can help me perform my research in a given time. The following diagram is the design approach of the project.



Figure 3: Design of the project

I started with data collection. The main idea was to look for medical data with many classes also which supports in building classification model. I choose medical data is because it is important to find out bias in such models. If bias is found, then it can lead to an incorrect diagnosis. During the long search for the dataset that is suitable for my project approach, I found a website called open ML [4]. I made use of the lymph dataset which had three unique classes that are fibrosis. Malign\_lymph and metastases. It is a place where you can find interesting datasets for analyses purpose.

During the literature review I came across scikit-learn, which is a free software machine learning library. One of the features of scikit-learn is it connects to open ML servers and fetches data from the website. Scikit learns dataset function was used to read data from open ML without the necessary of downloading huge datasets. This is my approach on how I found and read the dataset for my research work. I further went on to build a classification model using the SVM classification algorithm.

To install PYCM into the system. The installation process is simple for all the operating systems. For the successful installation of PYCM, we need to have the latest version of python in the system. We can download latest version of PYCM that PYCM version 3.3. Once that is done, we need to go to the path where it is installed, and type pip install -r requirements.txt and run python setup.py install. This will successfully install PYCM in your system. The installation process is explained in more details in the PYCM document [5].

Graphical user interface, text, application, email

Description automatically generated

Figure 4: Importing libraries and building classifier

The above picture shows the steps I took to load the data from open ml [4] and build a simple classification model. From sklearn I imported datasets to load data from open ml, imported train\_test\_split function to split that dataset into training and testing, imported fetch\_open\_ml function which connects to open ml server and loads the dataset. From sklearn I imported SVM algorithm and built a simple classifier.

Table

Description automatically generated

Figure 5: PYCM Confusion matrix

In figure 5 you can see I have used confusion matrix function from PYCM library. The confusion matrix is of predicted values verses actual values of three unique classes from lymph dataset. First step for approaching towards the results is to come up with a confusion matrix with many classes. By using confusion matrix function of PYCM library successfully produced a confusion matrix of many classes. The confusion matrix provides more insights on how the model performs and does it have bias in it.

Text

Description automatically generated with medium confidence

Figure 6: overall and individual class statistics

I have used accuracy metric to check the overall performance of the metric and filtered out individual class statistics. These are the following metric I have taken into consideration.

* Accuracy: It is number of correct predictions from overall prediction.
* Area under ROC curve: AUC is equivalent to the probability that a classifier will prioritize a random positive instance higher than random negative instance [13].
* Diagnostic odds ratio: it calculates the proportion of likelihood of positive outcome to the likelihood of negative outcomes.
* Table

  Description automatically generatedMathew’s correlation co-efficient (MCC): It is a confusion matrix method of calculating Pearson product moment.
* True positive rate (TPR): Also called as sensitivity, recall or probability of detection of few fields. The calculates the proportion of positive outcomes that are correctly identified.
* True negative rate (TNR): Also called as specificity. It calculates the negative outcomes that are correctly identified.
* False positive rate (FPR): It calculates the proportion of negatives that still produces positive results.
* False negative rate (FNR): It calculates the proportion of positives when it still produces negative results.

Output in figure 4 will be discussed in-depth in the findings and discussion section.

The visualization of the confusion matrix and confidence interval is also produced as an output.

Chart

Description automatically generated

Figure 7: PYCM Confusion matrix heatmap

Chart, line chart

Description automatically generated

Figure 8: Confidence interval graph

plot\_ci function plots confidence interval for each class in the lymph dataset. Confidence interval estimates the “margin of error” or to use the proper term, confidence interval, on our observation. A confidence interval tells us that at a given level of certainty if our scientific model is correct, the true value in the population will likely be in the range identified; the larger the confidence interval the less certain the observation will be. There are several different approaches to calculating confidence intervals [12].

1. FINDINGS and DISCUSSION

The highlight of the project was to detect bias in a multi-class classification model. As I mentioned earlier there are 5 different types of bias and I will be focusing on algorithmic bias in this project.

The main highlight of the findings is the multi-class confusion matrix. The confusion matrix in the figure gives more insights on how the classification has performed. Looking into the values that confusion matrix has produced, the model was very accurate at classifying metastases that is true positive/all=1.0. The accuracy for class malign\_lymph was good but not accurate that is 11/12=0.97 and for class fibrosis the accuracy is 0 cause it failed to predict correct instances. This clearly shows/ detects algorithmic bias in the multi-class classification model. This can lead to an incorrect diagnosis of a patient’s health condition.

A confidence interval gives insight into the margin of error. The confidence interval tells us that at a given level of certainty, if our scientific model is correct, the true value in the population will likely be in the range identified; the larger the confidence interval the less certain the observation will be. For class fibrosis, it is 0 because it completely failed to predict correct instances, for class metastases it has no interval cause the class had 100% accuracy and for malign class the confidence interval is between 0.35 to 0.80.

In figure 6 where it shows the overall statistics and individual class statistics of the classification model. The overall accuracy of the model is 81%. The model tends to learn some historical bias and fails to classify correctly. The AUC is equivalent to the probability that a classifier will prioritize a random positive instance higher than a random negative instance. In each class, the fibrosis had the lowest value cause it failed to classify any correct instances. DOR value for class malign\_lymph is 23.375 which means the likelihood of positive outcomes to the likelihood of negative outcomes is high in class malign\_lymph. The false negative value for fibrosis is 2, malign\_lymph is 8 and metastases is 0. The values clearly shows that there is for sure algorithmic bias in the classification model as it fails to classify correctly.

1. CONCLUSION AND FUTURE WORK

In this paper, I have showed that checking for algorithmic bias in machine learning models can be done in a flexible manner. The library PYCM described above is a self-sufficient library for post-classification model evaluation that supports many classes. I have presented theory, explained about open-source PYCM library along with examples and plots. Along the way, I have plotted confidence intervals to check the margin of error.

The source code of this project and the document can be found at https:// . More details on PYCM can be found at [5].

In the future, we can try to import the whole code into R and make use of confusR package for the visualization of prior and posterior odds and check of prior abundance in the classification model.

1. ACKNOWLEDGMENT

Work on this project was completed supported by open-source library PYCM.

1. REFERENCES
   1. Mark Voortman (2021) How AI can help understand bias. Retrieved from <https://www.othot.com/blog/ai-help-understand-bias>
   2. Sara A. Metwalli. 5 types of machine learning bias every data scientist should know. Retrieved form <https://towardsdatascience.com/5-types-of-machine-learning-bias-every-data-science-should-know-efab28041d3f>
   3. Council of Europe. Guidelines on facial recognition. (2021). Retrieved from [https://www.coe.int/en/web/portal/-/facial-recognition-strict-regulation-is-needed-to-prevent-human-rightsviolations](Council%20of%20Europe.%20Guidelines%20on%20facial%20recognition,%2028%20Jan%202021.%20URL%20https:/www.coe.int/en/web/portal/-/facial-recognition-strict-regulation-is-needed-to-prevent-human-rightsviolations-.%20%5bp1%5d)
   4. Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. OpenML: networked science in machine learning. SIGKDD Explorations 15(2), pp 49-60, 2013.  [https://www.openml.org/](%20https://www.openml.org/)
   5. Haghighi, S., Jasemi, M., Hessabi, S. and Zolanvari, A. (2018). PyCM: Multiclass confusion matrix library in Python. Journal of Open-Source Software, 3(25), p.729.<https://www.pycm.ir>
   6. Jason Brownlee (2016). What is a confusion matrix in machine-learning. <https://machinelearningmastery.com/confusion-matrix-machine-learning/>
   7. Jakub Wiśniewski, Przemysław Biecek (2021). fairmodels: A Flexible Tool for Bias Detection, Visualization, And Mitigation. Retrieved from <https://arxiv.org/pdf/2104.00507.pdf>
   8. R. K. E. Bellamy, K. Dey, M. Hind, S. C. Hoffman, S. Houde, K. Kannan, P. Lohia, J. Martino, S. Mehta, A. Mojsilovi´c, S. Nagar, K. N. Ramamurthy, J. Richards, D. Saha, P. Sattigeri, M. Singh, K. R.Varshney, and Y. Zhang. Ai fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. IBM Journal of Research and Development, (2019). Retrieved from <https://doi.org/10.1147/JRD.2019.2942287>. [p2, 10]
   9. P. Saleiro, B. Kuester, A. Stevens, A. Anisfeld, L. Hinkson, J. London, and R. Ghani. Aequitas: A bias and fairness audit toolkit, (2018). Retrieved from <https://arxiv.org/abs/1811.05577>. [p2]
   10. Lovell, D (2021). confusR Draft vignette <https://rpubs.com/DavidLovell/confusR-dev>
   11. S. Bird, M. Dudík, R. Edgar, B. Horn, R. Lutz, V. Milan, M. Sameki, H. Wallach, and K. Walker. Fairlearn: A toolkit for assessing and improving fairness in AI. (2020). Retrieved from <https://www.microsoft.com/en-us/research/publication/fairlearn-atoolkit-for-assessing-and-improving-fairness-in-ai/>
   12. Wikipedia. (2021). Confidence interval. Retrieved from, <https://en.wikipedia.org/wiki/Confidence_interval>
   13. D. M. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," (2020) <https://arxiv.org/ftp/arxiv/papers/2010/2010.16061.pdf>
   14. Beauxis-Aussalet, E., & Hardman, L. (2014). Visualization of confusion matrix for non-expert users. In *IEEE Conference on Visual Analytics Science and Technology (VAST)-Poster Proceedings*. <http://vis.cs.ucdavis.edu/vis2014papers/VIS_Conference/infovis/posters/beauxis-aussalet.pdf>
   15. Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2019). A survey on bias and fairness in machine learning. <https://arxiv.org/pdf/1908.09635.pdf>
   16. Branco, P., Torgo, L., & Ribeiro, R. P. (2017, May). Relevance-based evaluation metrics for multi-class imbalanced domains. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining* (pp. 698-710). Springer, Cham <https://repositorio.inesctec.pt/bitstream/123456789/4643/1/P-00M-PQ3.pdf>