ON THE MCC-F1 METRIC FOR BINARY CLASSIFICATION

XUAN LI

Abstract. This report is based on the paper "The MCC-F1 Curve: A Performance Evaluation Technique for Binary Classification" (Chang Cao, 2020). There are two sections in this report. The first section provides a comprehensive summary of the paper, discussing the key findings, contributions, limitations and future improvements. The second section includes an extensional study. We firstly investigates further on how severity of imbalance in the datasets affects different metrics. Then we extend to the discussion point on whether there is an optimal classifier across all datasets, in which we conduct simulations and comparisons using machine learning classifiers.

Contents

1. Summary of the Original Paper

1.1. **Problem Addressed.** The original paper (Chang Cao, 2020) builds on existing literature regarding evaluation metrics for binary classification problems. Among the commonly used metrics, limitations arise when dealing with imbalanced datasets. The authors discuss methods such as the **Receiver Operating Characteristic (ROC) curve** and **Concentrated ROC**, which can be overly optimistic about a classifier's

Date: September 8, 2024.

performance when facing negatively skewed datasets, i.e., a high ratio of negative values to true positives (Fawcett, 2006) (Swamidass et al., 2010). Additionally, methods like the **Precision-Recall (PR) curve** and **Precision-Recall-Gain curve** provide limited information when comparing classifier performance on positively skewed datasets or in situations with a high cost for false negatives (Swamidass et al., 2010) (Flach & Kull, 2015).

To address these limitations in evaluating classifier performance on imbalanced datasets, the paper introduces a new evaluation method: the **MCC-F1 curve** and **MCC-F1 metric**. This approach combines the normalized **Matthews Correlation Coefficient (MCC)** with the **F1 score** (Chicco & Jurman, 2020). The authors argue that the **MCC-F1** metric is well-suited for evaluating binary classification tasks, regardless of whether the dataset is balanced or imbalanced. Through simulations, the authors show that this new method can outperform commonly used metrics like **ROC** and **PR** on imbalanced datasets, overcoming the limitations of existing metrics.

1.2. **MCC-F1 Curve and MCC-F1 Metric.** The newly proposed metric for evaluating binary classifiers, **MCC-F1**, combines the following two metrics as its major components.

(1) **Matthews Correlation Coefficient (MCC)**, provides a comprehensive evaluation of all the elements in the confusion matrix and therefore can handle imbalanced datasets well.

$$MCC = \frac{1}{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}$$

The value of **MCC** ranges between $[-1, 1]$, where 1 represents a perfect classifier, 0 represents a random classification, and -1 represents a completely incorrect classifier. It is important to note that a high **MCC** value can only be obtained when both true positives and true negatives are high, and both false negatives and false positives are low.

(2) **Unit-normalized MCC**, a normalized version of the MCC, is used in both the MCC-F1 curve and the MCC-F1 metric calculation. The formula of **Unit-normalized MCC** is

$$\frac{\mathbf{MCC} + 1}{2}$$

The reason for normalization here is to maintain consistency with the **F1** score, which lies in the range $[0, 1]$.

(3) **F1 Score**, is the harmonic mean of precision and recall. It is calculated as:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

The F1 score emphasizes on the balance between precision and recall. It is a commonly used metric when dealing with imbalanced datasets. The value for F1 is between $[0, 1]$,

The **MCC-F1 curve** plots **unit-normalized MCC** on the Y-axis and **F1 score** on the X-axis, for different threshold values. This makes it easy to compare how classifiers perform regarding various threshold values. The point $(1, 1)$ is the perfect score; it is the best score that a threshold can possibly reach. An example of the **MCC-F1 curve** plot for a testing classifier with testing dataset and threshold values $\{0.1, 0.2, 0.3, 0.4, 0.5, 1.5\}$ is shown in Figure 1.2.

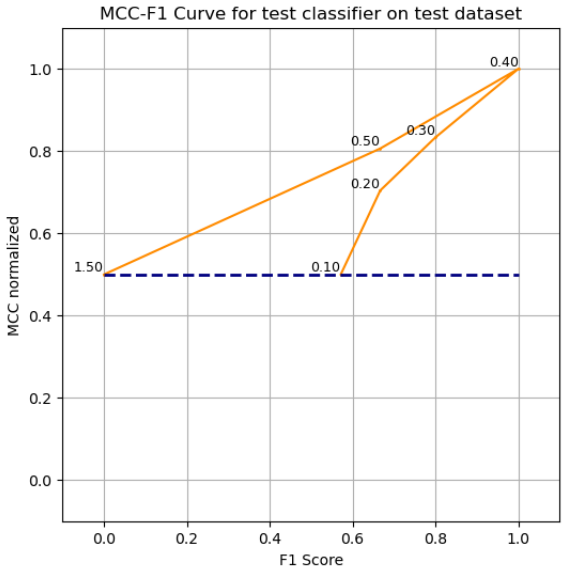

FIGURE 1. An example of the MCC-F1 curve, with the dashed line representing a random classifier.

To give an overall score for a classifier instead of just providing scores at each threshold, the paper introduces the **MCC-F1 metric**. This is a scale value that summarizes the classifier's overall performance across all threshold values. The process for calculating the MCC-F1 metric as outlined in the paper includes the following key steps:

1. **Dividing into left and right sides**:
   - **Left side**: threshold values bigger or equal to the best threshold value.
   - **Right side**: threshold values smaller than the best threshold value.

This division aims for a balanced evaluation across different threshold values and prediction scores, giving equal weight to both sides. Notice that we've slightly modified the definition from the original paper to make it clearer.

2. Sub-dividing MCC ranges into sub-ranges: The MCC-F1 curve is split into **100 sub-ranges** along the MCC axis:

$$\text{subrange size} = \frac{\max(\text{MCC}) - \min(\text{MCC})}{100}$$

This divides the MCC values into equal intervals. For each sub-range, we calculate the mean Euclidean distance from each point in the sub-range to the perfect performance point $(1, 1)$.

3. Calculating mean Euclidean distance: For each point (X_i, Y_i) on the MCC-F1 curve, compute the Euclidean distance to the perfect performance point:

$$D_i = \sqrt{(X_i - 1)^2 + (Y_i - 1)^2}$$

where X_i is the unit-normalized MCC, and Y_i is the F1 score.

4. Averaging over sub-ranges: For each sub-range, calculate the average distance of points within that sub-range j:

$$\bar{D}_j = \frac{\sum_{i \in \text{sub-range}} D_i}{n_j}$$

where n_j is the number of points in the sub-range j.

5. Grand average distance: The grand average distance D^* is calculated as the average of all sub-range mean distances across both left and right sides:

$$D^* = \frac{\sum_{(s,j)} \bar{D}_j}{|P|}$$

where P is the set of sub-ranges with non-zero points.

6. Final MCC-F1 metric: The final MCC-F1 metric is calculated as:

$$\text{MCC-F1 metric} = 1 - \frac{D^*}{\sqrt{2}}$$

This value ranges from 0 to 1, with a value of 1 indicating perfect performance.

This approach provides a balanced evaluation of classifier performance across different prediction thresholds while avoiding bias from uneven distribution of points. An example of the right-left sides and sub-range division plot is shown in Figure 1.2.

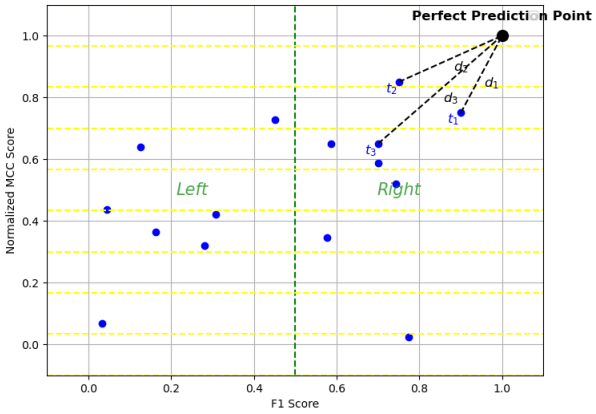

FIGURE 2. The green dashed line divides the points into left and right sides, and the yellow dashed lines divides the points into sub-ranges.

1.3. **Comparison to Other Metrics.** The paper compares MCC-F1 with ROC and PR through simulations under different datasets. The experiment includes 2 classifiers and 3 datasets. The 3 simulated datasets only differ in their ground truth values of 0 and 1. Since this is a binary classification setup, 0 represents the negative class and 1 represents the positive class. They are the only possible outcomes. Dataset X is skewed toward the negative class, with 1,000 positives and 10,000 negatives. Dataset Y is skewed toward the positive class, with 10,000 positives and 1,000 negatives. Dataset Z is perfectly balanced, containing 10,000 positives and 10,000 negatives.

The authors simulate prediction scores for two classifiers using different Beta distributions. For **classifier A**, they use a Beta(12, 2) distribution for the first 30% of positive cases and Beta(3, 4) for the remaining positives. This approach models a classifier with high recall at higher thresholds and lower recall at lower thresholds. For negative cases, they sample scores from Beta(2, 3). For **classifier B**, the scores for positive cases are all sampled from Beta(4, 3), and the negative cases are sampled from the Beta(2, 3).

Negatively Imbalanced Dataset: For this dataset, **classifier B** appears to perform better than **classifier A** in the ROC analysis (with an AUROC value of 0.73 for **classifier B** and 0.69 for **classifier A**). However, in the PR analysis, **classifier A** has a clear advantage when recall is below 0.4 but when recall exceeds 0.4, the difference between them is small. The overall AUPR value is 0.3 for **classifier A** and 0.2 for **classifier B**. In the MCC-F1 analysis, **classifier A** outperforms **classifier B**, with a better MCC-F1 curve and a higher score (0.35 for **classifier A** and 0.34 for **classifier B**). This result shows that **classifier B** does not perform as well as the

ROC analysis suggests. ROC curves can be misleading in cases with a big imbalance between negative and positive classes.

Positively Imbalanced Dataset: For this dataset, **classifier A** and **classifier B** appear to perform similarly (almost identical) in the PR analysis (with the same AUPR value of 0.96 for both **classifier A** and **classifier B**). However, in the ROC analysis, **classifier B** has a clear advantage over **classifier A** in the plot. And the overall AUROC value is 0.69 for **classifier A** and 0.73 for **classifier B**. In the MCC-F1 analysis, both classifiers appear to perform closely when F1 < 0.62 but when F1 exceeds 0.62, **classifier B** outperforms **classifier A** clearly. The overall MCC-F1 value is 0.49 for **classifier A** and 0.59 for **classifier B**. This result shows that PR is much less informative than AUC and MCC-F1 under positively imbalanced datasets.

Balanced Dataset: For this dataset, **classifier A** and **classifier B** appear to perform similarly in all three analysis (ROC, PR and MCC-F1). The plots show that the ROC, PR and MCC-F1 curves for two classifiers cross each other at some point. Both classifiers have the same AUPR value of 0.71. The AUROC value is 0.69 for **classifier A** and 0.73 for **classifier B**. The MCC-F1 value is 0.46 for **classifier A** and 0.53 for **classifier B**. This result shows that MCC-F1 metric has a bigger difference between the two classifiers. Therefore the MCC-F1 metric makes it more clear to select the best classifier than ROC and PR analysis.

In summary, MCC-F1 outperforms ROC and PR in many of the situations including the three cases discussed above. The **MCC-F1** analysis provides a more complete evaluation by considering all elements of the confusion matrix, making it particularly useful for imbalanced datasets as well as balanced ones. And it provides a straightforward way to select the best thresholds value.

1.4. **Highlights, Limitations and Future Improvements.** Imbalanced datasets are common in the real-world, especially in areas like medicine, genomics, and fraud detection. In these situations, the MCC-F1 curve provides a more accurate and fair way to assess classifier performance. This is a key contribution of the paper. Additionally, MCC-F1 curve presents a straightforward view to identify the best threshold value. Identifying the best threshold for a classifier is critical in real-world applications, especially when the cost of false positives and false negatives is unequal. The MCC-F1 metric also simplifies comparisons between classifiers by offering a single value, the MCC-F1 metric, which can summarize overall performance. While the MCC-F1 curve and metric provide clear advantages, there are some limitations.

One limitation of the MCC-F1 analysis in the original paper is the **lack of theoretical justification**. This includes elaboration on why combining MCC and F1 works as an effective metric. The paper introduces this combination and claims it works based on the nature of MCC and F1 metrics, but does not provide enough detail to explain why this metric is valid or superior, and how it is better than just using MCC or F1 metric alone. Additionally, the MCC-F1 metric formula **leaves some ambiguity**. For instance, the division of the left and right side in metric calculation is unclear. The original paper states that the division is based on the best prediction score (detail in page 7 of (Chang Cao, 2020)), but achieving a score better than the best score is not possible. This would result in the left side having only a single point, which is the threshold with the best prediction score. Furthermore, in extreme cases where one side is empty, the paper does not specify how to handle that. Should we simply assign zero for the missing side and divide to get the average, or rely solely on the available side's value? The paper also lacks guidance on the optimal number of sub-ranges division for balancing accuracy and computational cost, making it unclear why dividing into 100 sub-ranges adds value. Dividing into too many sub-ranges, when fewer threshold values exist, seems excessive and unnecessary. These gaps in the explanation create uncertainty on how to apply the metric in practice.

One limitation of the MCC-F1 analysis is the **complexity of interpretation**. While it combines two important metrics, the complexity of the structure also increases. In fact, MCC itself is often viewed as harder to interpret than more straightforward metrics like AUC and PR. For practitioners who preferred interpretable methods, the complexity in the MCC-F1 metric potentially requires additional explanation and interpretation of its outcome. Another challenge is its **scalability to extend to multi-class problems**. The paper focuses solely on binary classification setting. Adapting the MCC-F1 curve to multi-class classification setting would require modifications for the re-structuring of the metric. Since this metric requires more computation than just comparing existing evaluations, it also adds complexity when extending to a broader range of use cases. Lastly, the paper's findings are heavily based on **simulation data**. How well the MCC-F1 curve performs in real-world datasets remains unknown. We propose several directions for future improvements and further analysis.

(1) We can explore different left-right division methods, such as using the mean or median F1 score. Since the current method relies on MCC for sub-range division, similar approaches using F1 score could be tested. In a quick simulation, we found that different division methods, like median and mean F1

scores, resulted in more balanced outcomes depending on the specific dataset and classifier. How to choose the most appropriate division when calculating

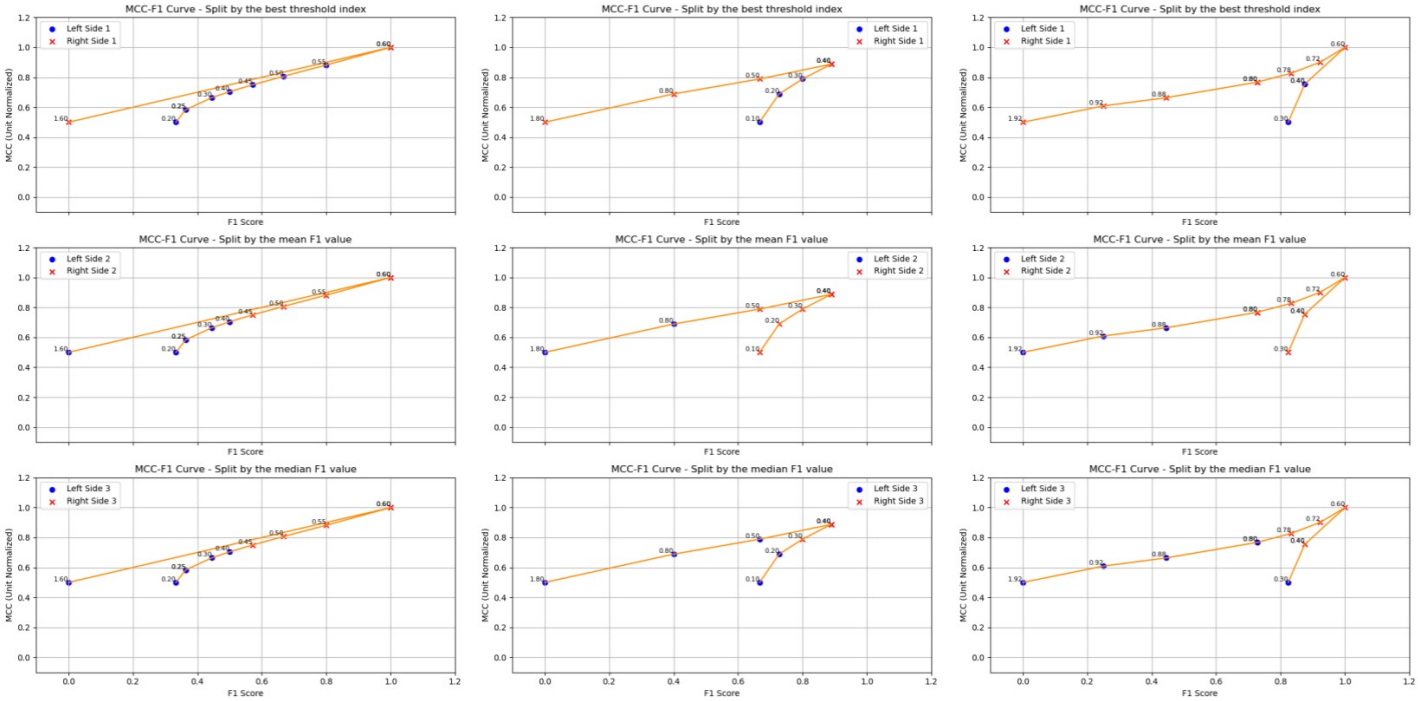

FIGURE 3. Different divisions for (1) Negatively skewed dataset, (2) balanced dataset, and (3) positively skewed dataset

(2) We can extend the analysis to multi-class classification with some adjustments to the formula. Alternative approaches to structure the multi-class analysis should be considered. For instance, the confusion matrix can be an $N \times N$ matrix. Each row of the matrix represents the actual class (true labels), and each column represents the predicted class. Alternatively, we can simplify the multi-class confusion matrix by just treating it as a binary classification problem (i.e., treating each prediction as either correct or incorrect, without distinguishing between specific classes).

(3) **Proposed by Prof. Keegan Korthauer**: A deeper analysis is needed to explore whether the severity of imbalance affects certain metrics more than others. This can be examined both analytically and through simulations.

(4) **Proposed by Prof. Keegan Korthauer**: A key discussion point is whether the balance between positive and negative classes should influence the choice of optimal classifier. Is it possible for one classifier to universally outperform others?

(5) Finally, we propose comparing the performance of the MCC-F1 metric not only against ROC and PR analyses but also against other methods like MCC,

F1, Balanced Accuracy, G-Mean (Geometric Mean), Fowlkes-Mallows Index (FMI), and so on. One can apply and compare these methods on real-world datasets to better understand their effectiveness in practical scenarios.

2. Extensional analysis

In this section, we explore the topics on items **3** and **4** from the list of proposed future improvements. We address these two discussion points in separate subsections. For each, we set up simulations in Python for our investigation.

2.1. **Severity of imbalance in different metrics.** We explore how the severity of imbalance affects different metrics. Specifically, we focus on a limited set of metrics: **ROC** (**AUC ROC** in the following figures stands for **Area Under Curve for ROC**), **PR** (**AUC PR** stands for **Area Under Curve for PR**), and **MCC-F1**. In the first simulation experiment, we generate datasets ranging from negatively skewed to positively skewed. Our approach adjusts the ratio between the number of positives and negatives, while keeping the total number of data points fixed at 10,000 for each dataset. The ratio is adjusted from 1:49 to 49:1 over 13 datasets. This means that in the first dataset, we have 200 positives and 9,800 negatives. We adjust the ratio gradually. For the 7th dataset, we have a perfectly balanced dataset with 5,000 positives and 5,000 negatives, and for the 13th dataset, we have 9,800 positives and 200 negatives. Results are presented in Figure 4. We applied two classifiers to the datasets, which are generated using the same processes as in the original paper, employing beta distributions.

Our simulation shows that as the ratio between positives and negatives changes, both the **MCC-F1 metric** and **AUPR** change accordingly. There is a positive relationship between the ratio and these metrics: as the number of positives increases and the number of negatives decreases, the **MCC-F1 metric** and **AUPR** improve. In contrast, the **AUROC** remains relatively stationary across different ratios for both classifiers.

When comparing the two classifiers, **AUROC** consistently shows that **classifier B** outperforms **classifier A**. **MCC-F1** follows this trend, except for the first several extremely negatively skewed datasets, where classifier A performs better. **AUPR**, however, initially favors **classifier A** but gradually overlaps with **classifier B** as the dataset becomes more positively skewed. This aligns with the authors' observation that **PR** curve can be less informative for classifier comparison on positively skewed

datasets. In our case, the overlap occurs just after the balanced dataset.

Given how we build the classifiers, it makes sense why **classifier B** performs better overall. **Classifier A** generates 70% of positives using a beta(3,4) distribution, while **classifier B** generates 100% of positives using beta(4,3). As the dataset becomes more dominated by positives, **classifier B**'s advantage on identify positives leads to an overall better performance.

In the second simulation experiment, we perform a relative sensitivity analysis for **MCC-F1 metric**, **AUROC**, and **AUPR**. We plot the percentage changes in classifier performance relative to the perfectly balanced dataset. We use the following formula:

$$metricPctChange = \frac{metricScore_i - metricScore_b}{metricScore_b}$$

where $metricScore_i$ is the score of the metric for dataset i, and $metricScore_b$ is the score of the metric for the perfectly balanced dataset. The percentage change values are rounded to two decimal places. Results are plotted in Figure 5.

As shown in the plots, the percentage change for **AUROC** remains quite stable for both classifiers. This indicates that, regardless of the class imbalance in the dataset, the **AUROC** does not vary much for the same classifier. In contrast, the percentage changes for both **AUPR** and the **MCC-F1 metric** are much more pronounced. These metrics are sensitive to class imbalance. However, the percentage changes for **AUPR** are very similar across classifiers, while the percentage changes for the **MCC-F1 metric** show more noticeable differences. Results are shown in figure 5.

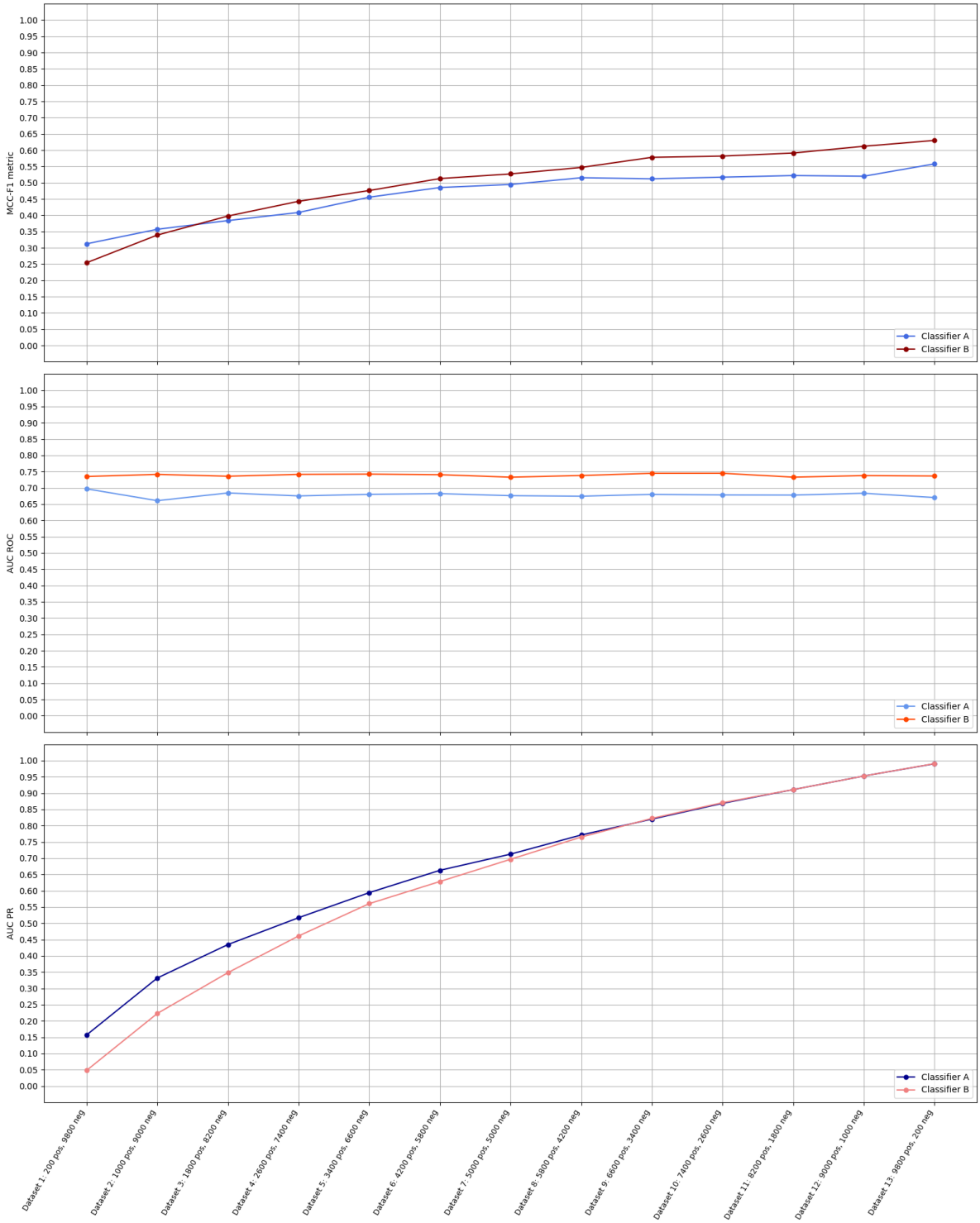

FIGURE 4. Metric values with respect to variations in datasets of different positives: negative ratios for (1) MCC-F1 metric, (2) the area under curve for ROC, and (3) the area under curve for PR

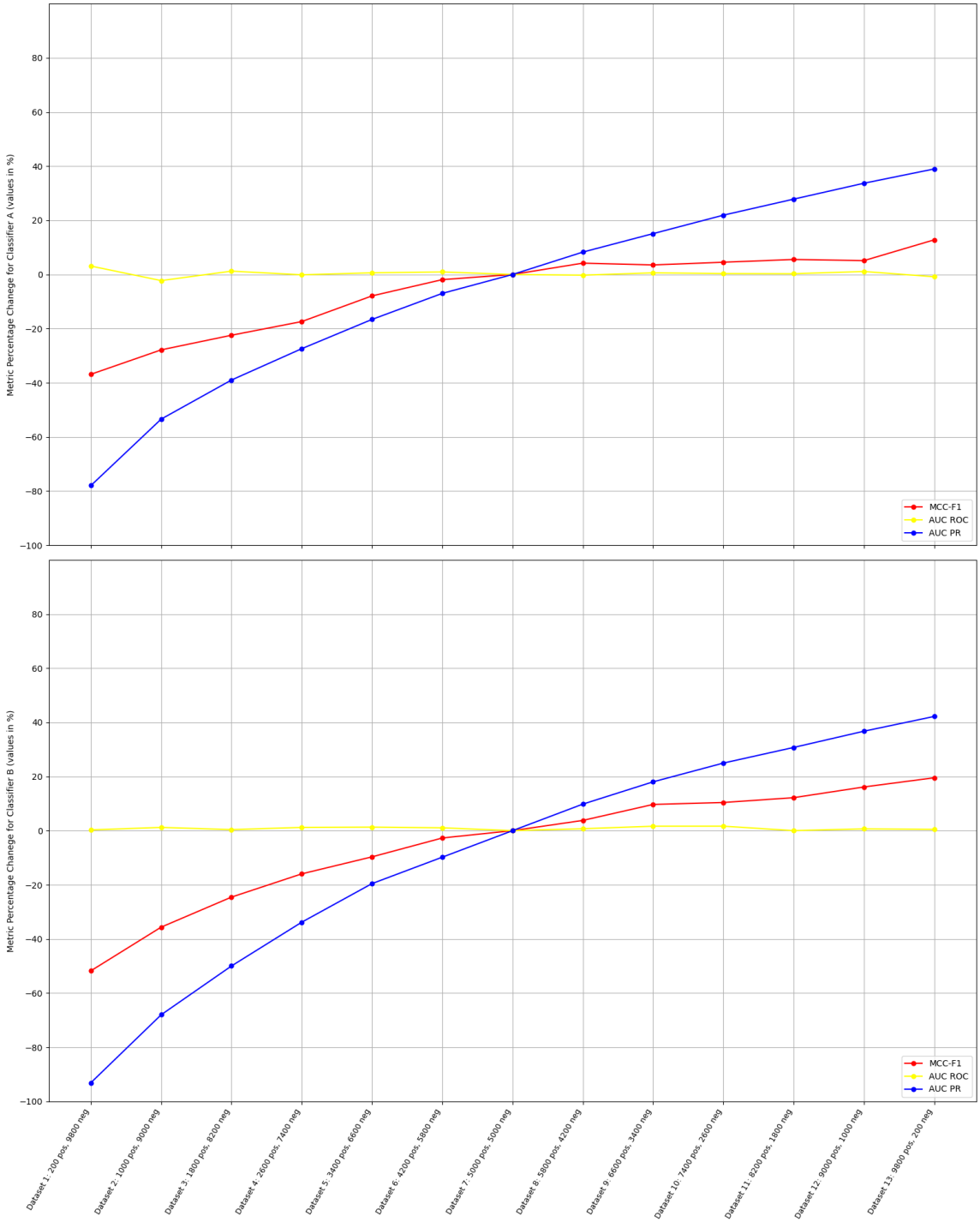

FIGURE 5. Performance Percentage Change: the top plot shows the percentage change of 3 metrics for classifier A compared to the perfectly balanced dataset and similarly for classifier B at the bottom plot. The percentage change values at 7th dataset (in the middle) for all metrics and classifiers should always be 0.

To investigate further, we conduct experiment 3. In this set up, we don't use any specific classifier but simply plot the different metrics as each element in the confusion matrix (TP, TN, FP, FN) varies. In each plot, we hold three elements constant and vary one. We create four plots in total, allowing us to examine whether certain metrics are sensitive to changes in confusion matrix elements. The datasets are not kept at a consistent size to allow for imbalanced scenarios. The results are shown in Figure 6. In this experiment, we break down the **MCC-F1**, **ROC**, and **PR** metrics by plotting **MCC**, **F1 score**, **precision**, **recall**, and **FPR** separately. The definitions are given below:

$$\text{Precision} = \frac{TP}{TP+FP}, \text{Recall} = \frac{TP}{TP+FN}$$

$$\text{FPR} = \frac{FP}{FP+TN}, \text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{MCC} = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

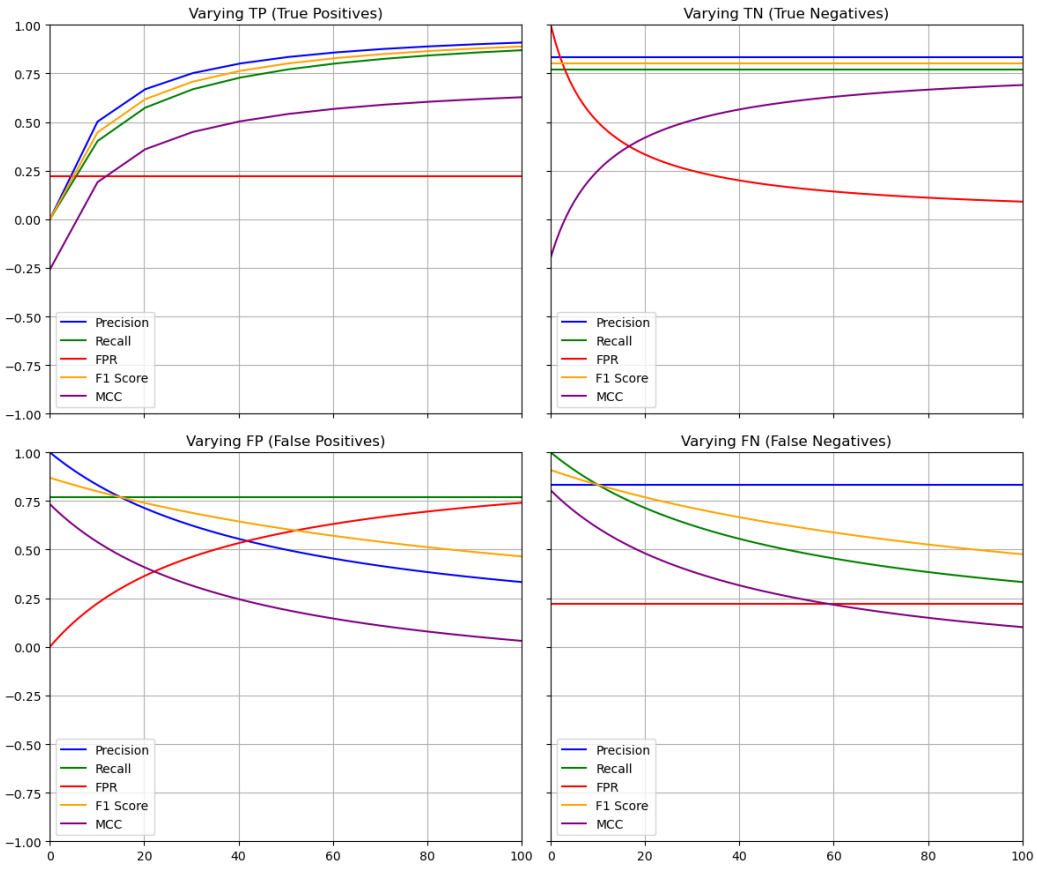

FIGURE 6

As the results indicate, **FPR** is not sensitive to changes in *TP* and *FN*, while **precision**, **recall**, and **F1 score** are not sensitive to changes in *TN*. Additionally, **recall** is not sensitive to changes in *FP*, and **precision** and **FPR** are not sensitive to changes in *FN*. These findings can also be inferred from the formulas that compose these metrics.

Due to the structure of the **ROC** curve, which plots **recall** against **FPR**, it makes sense why **ROC** does not perform well with negatively skewed datasets. In imbalanced datasets, especially when *TN* greatly outnumbers *TP*, even a rise in *FP* may not significantly affect **FPR** due to the large number of negatives. Moreover, the classifier may increase *TP* by lowering the threshold. This also increases *FP*, but can be offset by the large number of *TN*. If the classifier correctly identifies most positives, the **recall** may appear high, which can be misleading when combined with the low **FPR**.

In contrast, the **PR** curve performs better in negatively skewed datasets because it emphasizes the classifier's ability to predict the positive (minority) class. However, the **PR** curve loses its effectiveness in positively imbalanced datasets. With few negative instances, *FP* has less impact on precision, and *FN* has less impact on recall. Even if a classifier makes several *FP*s, precision will not drop significantly because the total number of positives is large. Similarly, recall remains high as the number of *FN* is small. Consequently, the **PR** curve becomes less informative in distinguishing between classifiers.

The **F1 score** faces similar advantages and challenges in both negatively and positively imbalanced datasets since it focuses on the positive class. However, **MCC** is sensitive to mis-classifications in both classes. It only achieves a high value when both *TP* and *TN* are high, and *FP* and *FN* are low. Therefore it provides a more comprehensive assessment for classifier performance across both the majority and minority classes. Overall, **MCC-F1** analysis offers more meaningful insights than **ROC** or **PR** analysis.

It's important to note that the choice of metrics is often depended on the specific task and its purpose. The **F1 score** (with $\beta = 1$) finds a balance between Precision and Recall, making it useful when both false positives and false negatives are equally important in the analysis task. However, changing the β value in the F-score can shift that balance. A higher β (like **F2**) would place more weight on the Recall, which

is ideal when missing positives is more costly. On the other hand, a lower β (such as **F0.5**) would place more weight on the Precision, which is helpful when avoiding false positives is more critical. One can incorporate different F-scores in the MCC-F1 analysis to bring out different strengths and weaknesses of classifiers according to the problem setting.

2.2. **Optimal classifier across datasets.** A summary table for the classifiers performances in the previous simulation experiment 1 is presented in Figure 7. Although in simulation 1, **classifier B** appears to perform better almost always than **classifier A**, we still observe cases when **classifier A** performs better than **classifier B**. This happens on the extremely negatively imbalanced datasets. Notice that how we constructed **classifier B** has an intuitive advantage over **classifier A** since we make a bigger difference in the positive class prediction scores for **classifier B**. Even under this "unfair" construction of the classifiers, we still cannot state that **classifier B** performs ultimately better under all circumstances. If we were to use the two classifiers on a target dataset that is extremely negatively imbalanced, say the ratio between positives and negatives is 1:100. As observing the current trend in the simulation, we cannot conclude that **classifier B** will perform better in this case.

	Datasets	Classifier A wins # times	Classifier B wins # times	Average for A - B
Negative Skewed datasets	1 to 4	2	2	0.006875
Balanced datasets	5 to 9	0	5	-0.035600
Positive Skewed datasets	10 to 13	0	4	-0.074525
Total Datasets	1 to 13	2	11	-0.034508

FIGURE 7. Performance comparison between classifier A and classifier B: the winner for different types of datasets is highlighted in blue

To elaborate on a more general context, we conduct experiment 4. In this simulation, we evaluate the performance of various machine learning classifiers across datasets with different levels of imbalance using the **MCC-F1** metric. We use five classifiers: **Logistic Regression**, **Support Vector Machine (SVM)**, **K-Nearest Neighbors (KNN)**, and **Naive Bayes**. For each classifier, we test its performance on 3 datasets: a balanced dataset (50% positive, 50% negative), a negatively imbalanced dataset (99% negative, 1% positive), and a positively imbalanced dataset (1% negative, 99% positive). Each dataset is generated using the *make_classification* function in the Python *SKLearn* package. We control the datasets to be with 10,000 samples and 20 features. Each dataset is split into training and testing sets (with 30% of data). After trained on the training set, each classifier outputs the prediction scores for the testing set. We then compute the **MCC-F1** metric on the testing set

for evaluating the classifier's performance.

Finally, the results for all different classifiers and datasets are plotted in Figure 8, where the **MCC-F1** metric value is plotted against the proportion of the number of negatives in the dataset. Comparing the classifier's performances across datasets, we see different classifier "winners". For the negatively imbalanced dataset, the **Logistic Regression** classifier outperforms other classifiers only slightly, with **Naive Bayes** classifier almost performs as good. For the balanced dataset, the **KNN** classifier outperforms other classifiers distinctively. And for the positively imbalanced dataset, the **Logistic Regression** classifier outperforms other classifiers moderately. This simulation demonstrates that the balance between positive and negative classes greatly influences the result of selecting a best classifier. There is no classifier that can universally outperforms others across different types of datasets. This is because that performance indeed depends on the class balance of the dataset and the evaluation metrics being used. A classifiers could obtain a very high performance score on imbalanced datasets, even if they simply predicts the majority class and ignores the minority class.

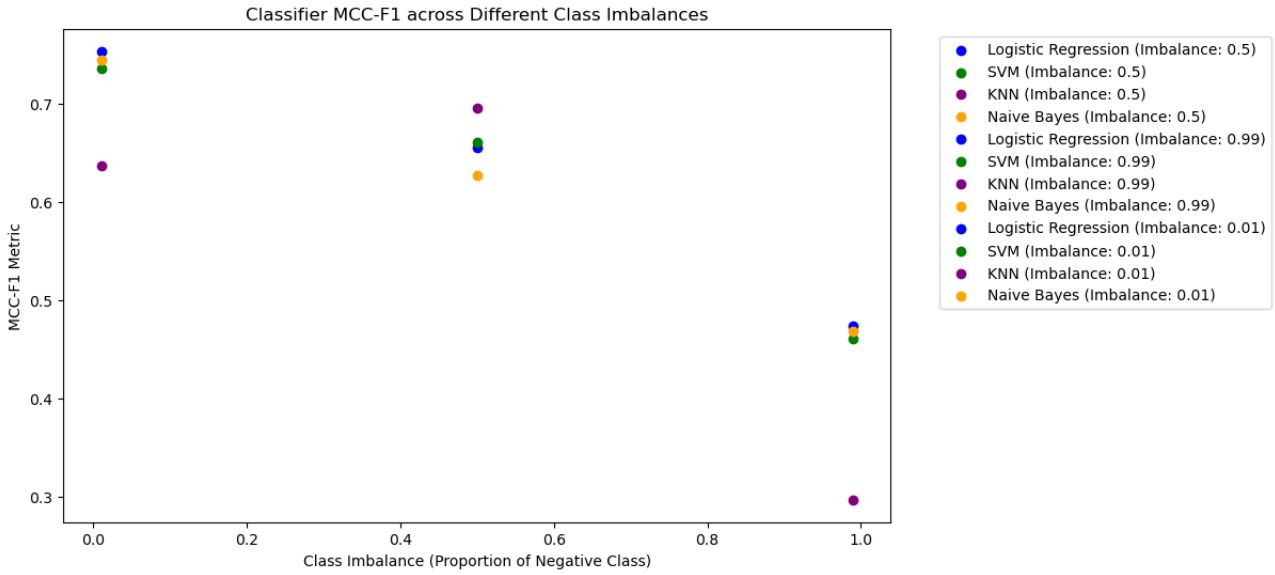

FIGURE 8

2.3. **Summary.** In this extensional study, we explore beyond the original paper for the effectiveness of the MCC-F1 metric in contrast to other metrics (ROC and PR). Through a variety of experiment simulations, we highlight MCC-F1's advantage in imbalanced classification problems. However, the most suitable metric always depends on the specific goal of the task. We extend our exploration to the discussion

point on whether the balance between positive and negative classes affects the choice for optimal classifier. In other words, we want to know if there is a single classifier that can consistently outperform others, regardless of the class distribution. Based on our findings, the answer is no. Since class balance does influence the classifier's performance substantially, there is no single classifier that can universally outperform across all datasets.

References

Chang Cao, M. M. H., Davide Chicco. (2020). The mcc-f1 curve: A performance evaluation technique for binary classification. *arXiv*, *stat.ML*(2006.11278).

Chicco, D., & Jurman, G. (2020). The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classifcation evaluation. *BMC Genomics*, *21*(1).

Fawcett, T. (2006). An introduction to roc analysis. *Pattern Recognition Letters, 27*(8), 861–874.

Flach, P., & Kull, M. (2015). Precision-recall-gain curves: Pr analysis done right. *Advances in Neural Information Processing Systems*, 838–846.

Swamidass, S. J., Azencott, C.-A., Daily, K., & Baldi, P. (2010). A croc stronger than roc: Measuring, visualizing and optimizing early retrieval. *Bioinformatics*, *26*(10), 1348–56.