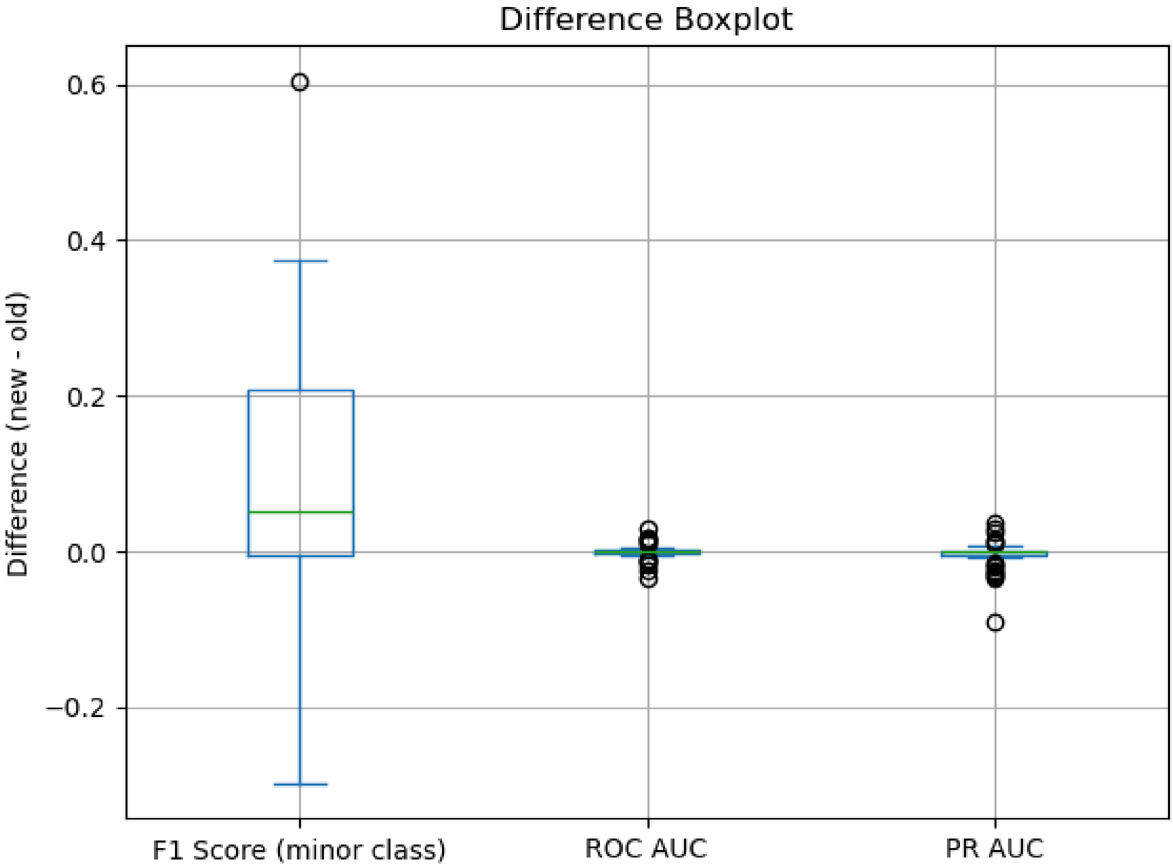
Interpretation of the results obtained with the changes implemented

Only using Class Weights

The weighting technique alone is very effective in increasing the F1 Score of the minority class, with statistically significant gains.
Both the ROC AUC and PR AUC were not significantly impacted.
Applying class weights proved to be effective in addressing class imbalance, leading to a significant and consistent increase in the minority class F1 score, without harming other performance metrics.

```
=== Difference Mean ===
F1 Score (minor class) 0.097604
ROC AUC                0.000033
PR AUC                 -0.003217
dtype: float64


=== Difference Median ===
F1 Score (minor class) 0.051273
ROC AUC                0.000064
PR AUC                 0.000000
dtype: float64


=== Difference Std Deviation ===
F1 Score (minor class) 0.176437
ROC AUC                0.009678
PR AUC                 0.018234
dtype: float64


=== N° of cases that the new algorithm as better results ===
F1 Score (minor class) 31
ROC AUC                25
PR AUC                 23
dtype: int64


=== N° of cases that the new  algorithm has worst results ===
F1 Score (minor class) 15
ROC AUC                20
PR AUC                 23
dtype: int64


=== Paired t-test (p-value) ===
```

```
F1 Score (minor class): p = 0.0003
ROC AUC: p = 0.9809
PR AUC: p = 0.2181
```

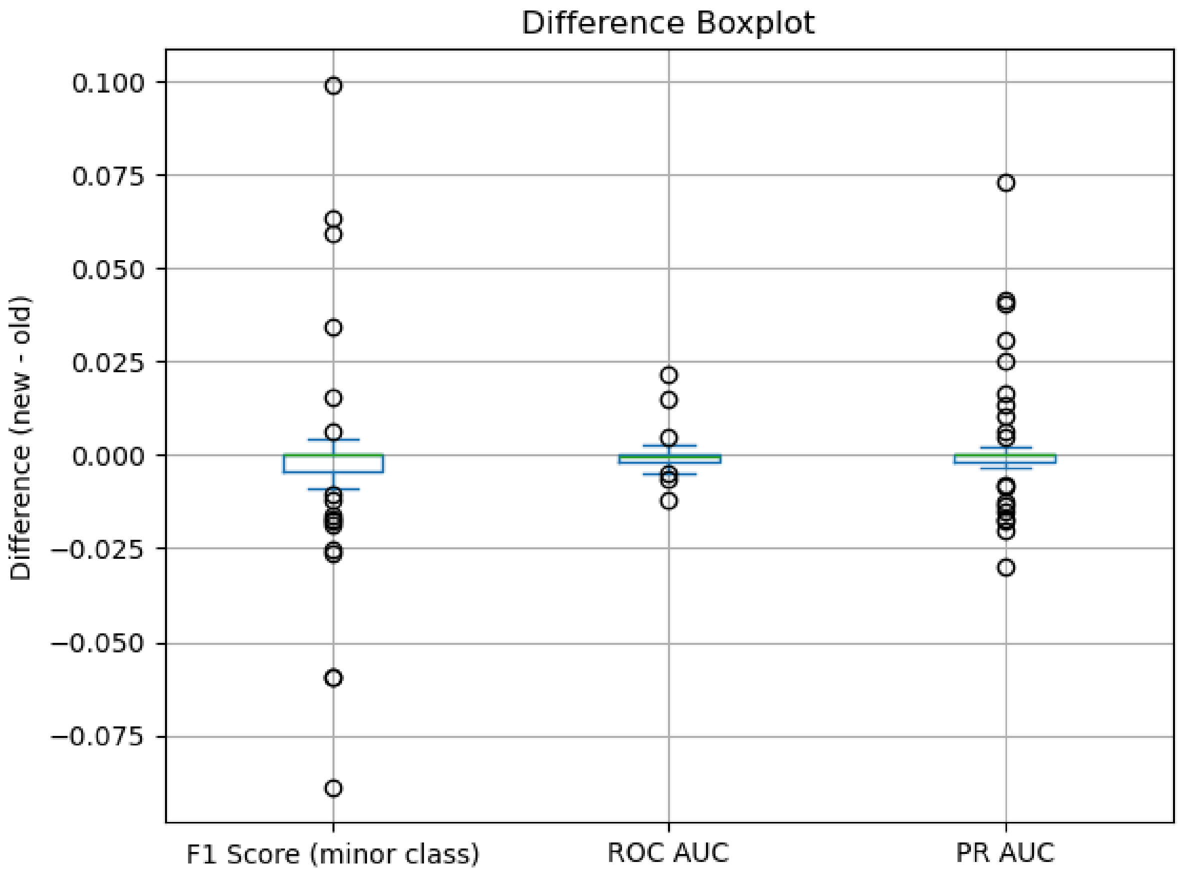


Only using Hellinger Gain(first tree)

It brought no statistical benefits in any metric.
It may even have slightly worsened the F1 Score.
The gains in PR AUC were small and statistically irrelevant.
The use of Hellinger gain alone did not lead to clear or significant improvements. While conceptually appealing, its isolated effect was limited in practice.

```
=== Difference Mean ===
F1 Score (minor class)    -0.001895
ROC AUC                  -0.000321
PR AUC                   0.002212
dtype: float64


=== Difference Median ===
F1 Score (minor class) 0.000000
```

```
ROC AUC                -0.000074
PR AUC                 0.000000
dtype: float64


=== Difference Std Deviation ===
F1 Score (minor class) 0.027326
ROC AUC                0.004681
PR AUC                 0.016516
dtype: float64


=== N° of cases that the new algorithm as better results ===
F1 Score (minor class)      9
ROC AUC                15
PR AUC                 22
dtype: int64


=== N° of cases that the new  algorithm has worst results ===
F1 Score (minor class) 20
ROC AUC                27
PR AUC                 21
dtype: int64


=== Paired t-test (p-value) ===
F1 Score (minor class): p = 0.6261
ROC AUC: p = 0.6295
PR AUC: p = 0.3483
```



Using both methods

The new technique is clearly beneficial for the F1 Score of the minority class, with statistically significant gains.

The ROC AUC and PR AUC metrics were not significantly affected, suggesting that the F1 Score gain came without seriously compromising overall performance.

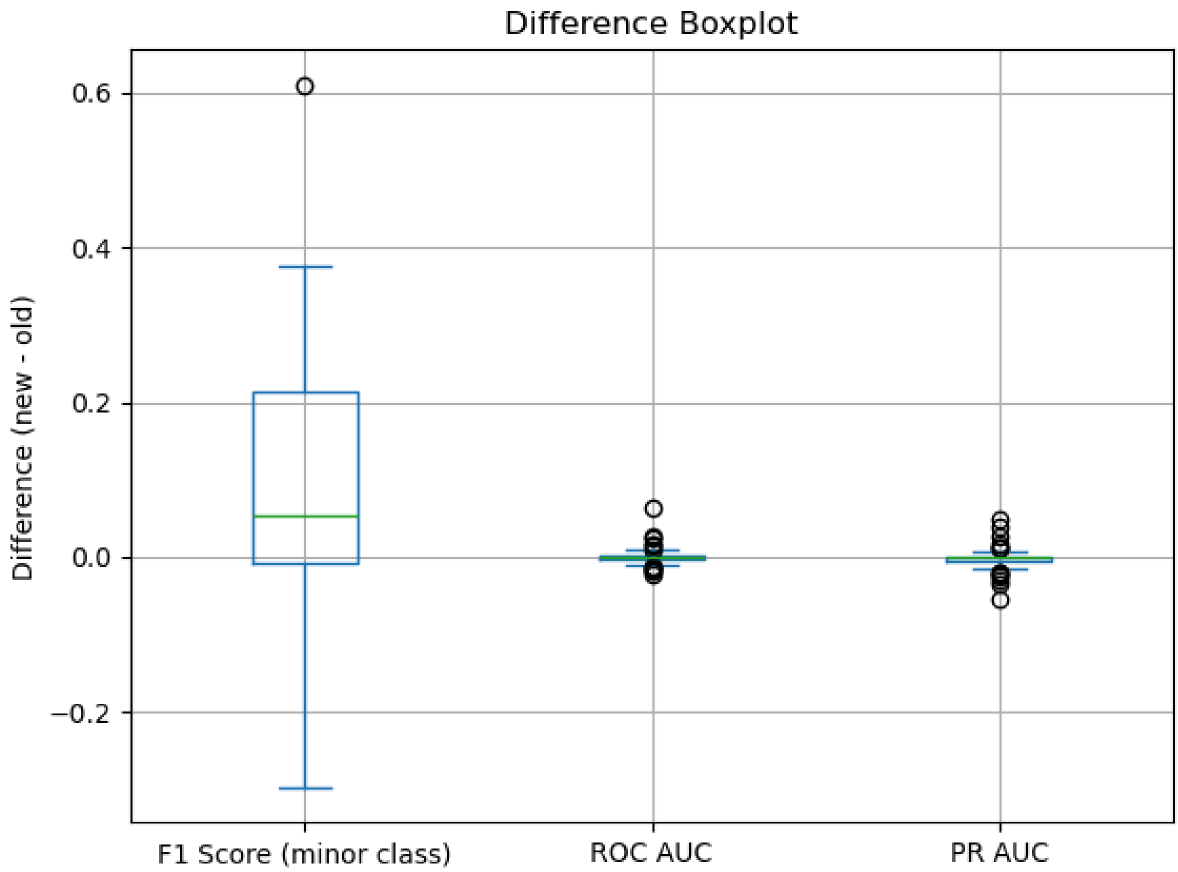
The focus on improving performance in the minority class was successful, with no serious side effects on the other metrics.

This approach is justified and statistically validated.

```
=== Difference Mean ===
F1 Score (minor class) 0.098198
ROC AUC                0.001146
PR AUC                 -0.001992
dtype: float64
```

```
=== Difference Median ===
F1 Score (minor class) 0.054082
ROC AUC                0.000074
PR AUC                 -0.000053
dtype: float64
```

```
=== Difference Std Deviation ===
F1 Score (minor class) 0.178420
ROC AUC                0.012614
PR AUC                 0.016229
dtype: float64


=== N° of cases that the new algorithm as better results ===
F1 Score (minor class) 31
ROC AUC                25
PR AUC                 22
dtype: int64


=== N° of cases that the new  algorithm has worst results ===
F1 Score (minor class) 15
ROC AUC                21
PR AUC                 25
dtype: int64


=== Paired t-test (p-value) ===
F1 Score (minor class): p = 0.0003
ROC AUC: p = 0.5235
PR AUC: p = 0.3898
```



Comparison with Scikit_Learn’s Gradient Boosting Classifier

(new algorithm -> our version of the algorithm using class weights and hellinger gain)

The improved model offers better minority class recognition, as evidenced by increased F1 scores in a majority of cases. However, this comes with a trade-off in overall discrimination and ranking ability, as seen in the significantly lower ROC AUC and PR AUC values. This highlights a common trade-off in imbalanced classification: improving recall and F1 for the minority class may lead to lower performance in metrics that consider the full probability distribution.

```
=== Difference Mean ===
F1 Score (minor class) 0.039902
ROC AUC                -0.257929
PR AUC                 -0.299602
dtype: float64

=== Difference Median ===
F1 Score (minor class) 0.024145
ROC AUC                -0.003368
PR AUC                 -0.021402
dtype: float64

=== Difference Std Deviation ===
F1 Score (minor class) 0.146112
ROC AUC                0.400393
PR AUC                 0.433906
dtype: float64

=== N° of cases that the new algorithm as better results ===
F1 Score (minor class) 31
ROC AUC                20
PR AUC                 19
dtype: int64

=== N° of cases that the new  algorithm has worst results ===
F1 Score (minor class) 18
ROC AUC                29
PR AUC                 30
dtype: int64

=== Paired t-test (p-value) ===
F1 Score (minor class): p = 0.0593
```

```
ROC AUC: p = 0.0000
PR AUC: p = 0.0000
```