Experiment

Creator **Olusanmi Hundogan** Created **Feb 29, 2024, 22:12** Last updated **Mar 06, 2024, 11:56**

Files

- experiment_bucket_classifiers.py
- analyze_bucket_classifiers.py

Motivation

The MMA model uses a two-step approach to prediction. The model trains a VotingRegression model for different buckets of scope values. For this, we need a reliable method to classify an expected bucket of a datapoint. We solve this by also training a classification model before prediction. Preliminary studies have shown that the chosen classifier performance determines the model regression performance. In this experiment, we investigate the interplay between time and performance of different classifier options.

The classifiers tested were:

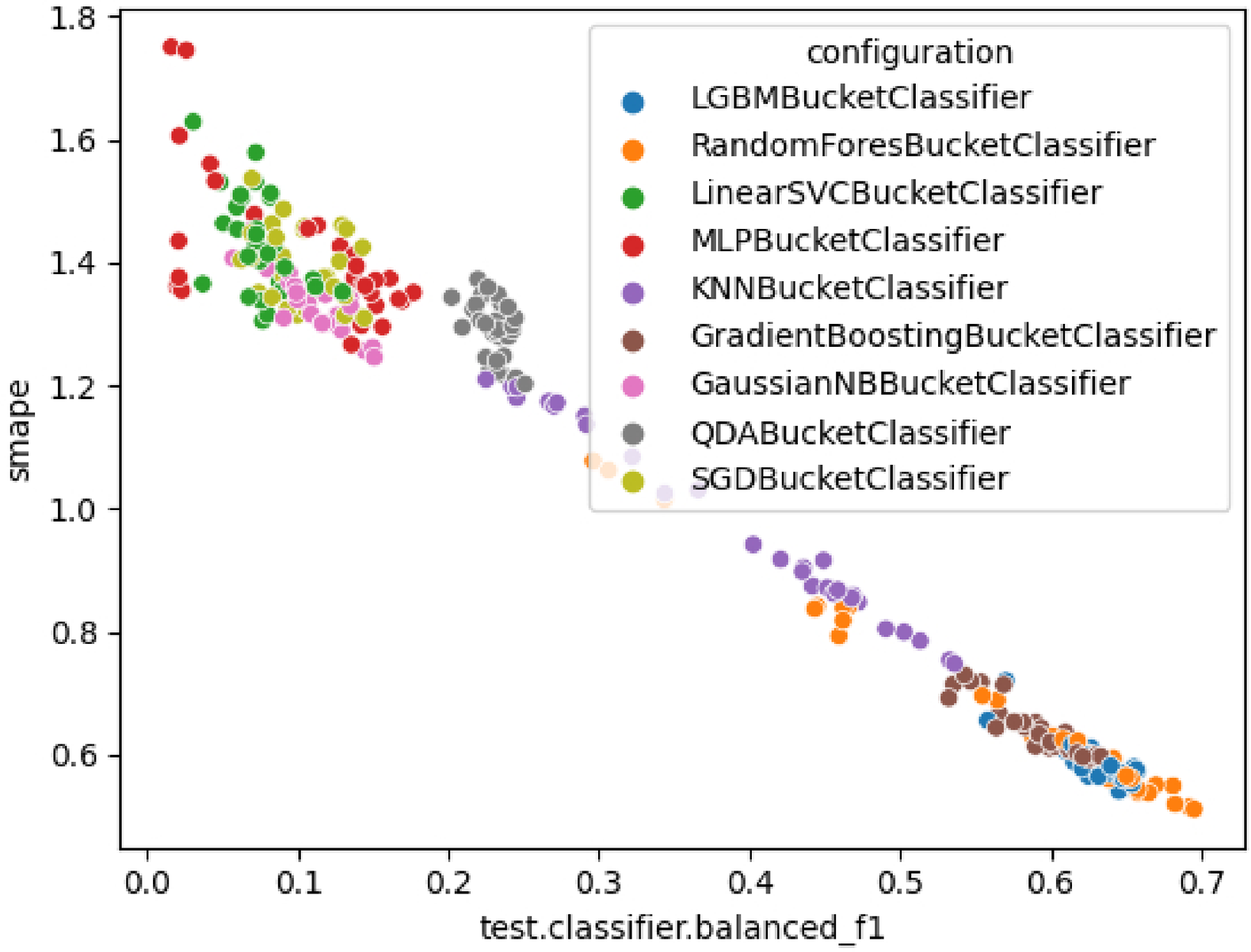
- LGBM
- RandomForest
- LinearSVC
- Multi-Layer-Perceptron
- KNN
- GradientBoosting
- GaussianNaive Bayes
- Quadratic Discriminant Analysis
- Stochastic Gradient Descent

Design

We implemented bucket classifiers for each classification method mentioned above and compared their performance in terms of balanced F1 scores, sMAPE and runtime. For the experiment, we kept the configuration of the pipeline as close to the baseline as possible in order to observe the direct effects of the classifiers on the final predictions instead of the effect of other components. We ran it for 10 iterations x 9 classifiers x 3 target scalers, only for scope 1 (since that was usually sufficient to provide the best insights).

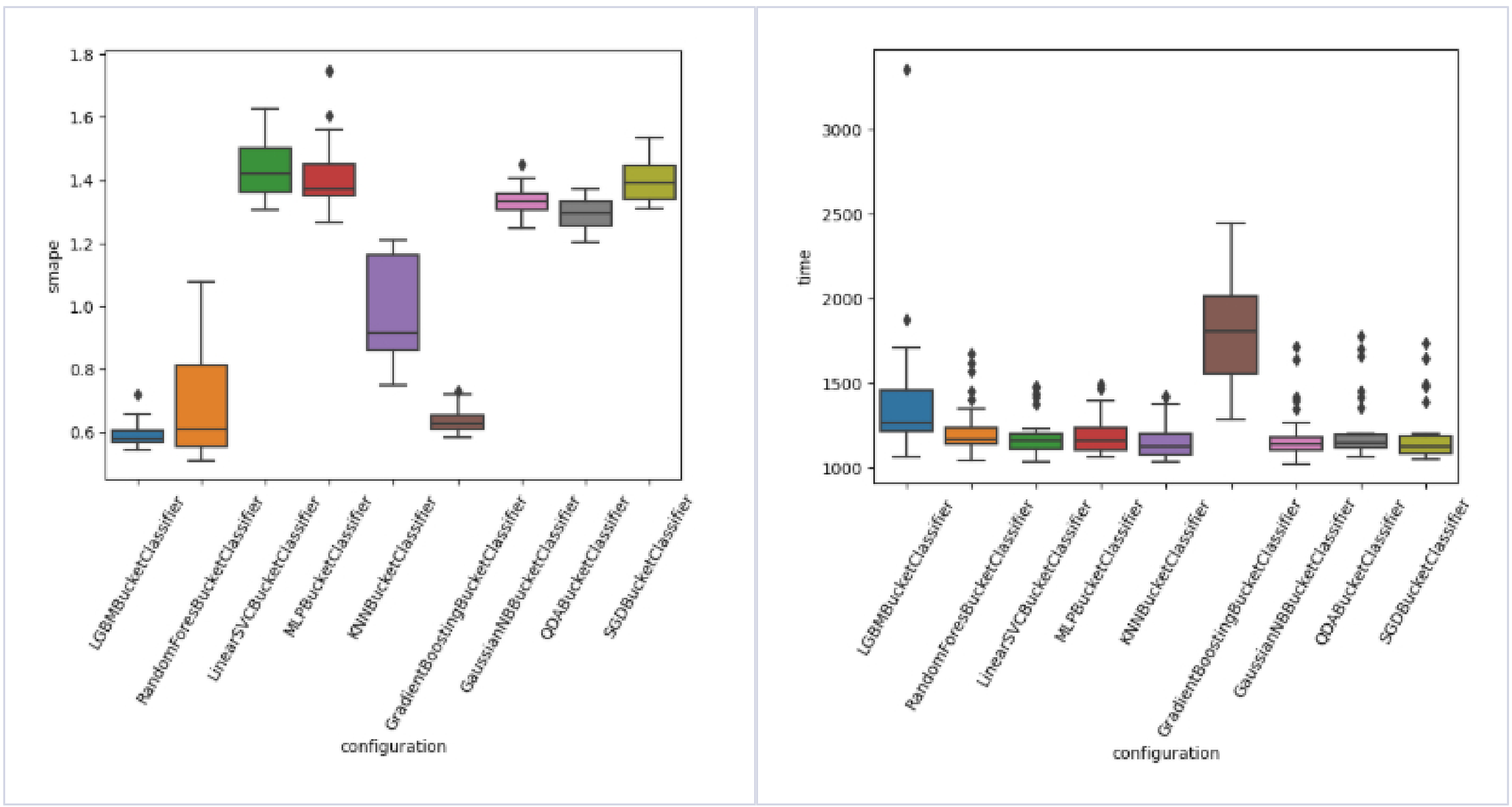
Results and Insights

The scatter plot reinforces the notion that the classifier performance is directly linked to the model performance as an increase in F1 leads to a decrease in sMAPE.

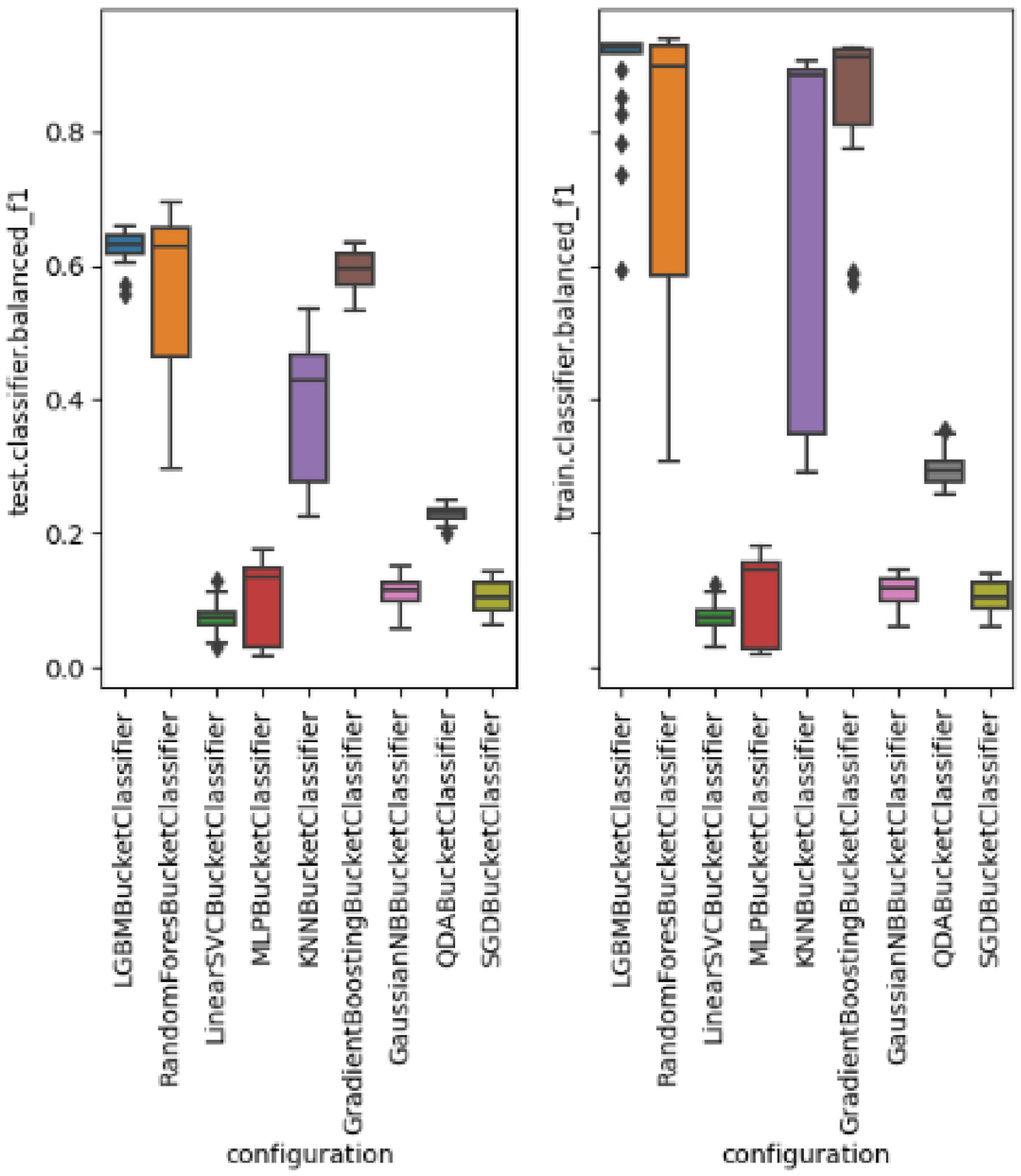


A remarkable result is that the Random Forest classifier performed better than LGBM, showing the lowest sMAPE, highest F1 score and a slightly improved runtime value. We can observe that its accuracy varies more than LGBM's, but this might be caused by the low number of iterations (10).

Another honorable mention is KNN, which scored very close to the previous two, despite its robustness and simplicity. It would have been a very good candidate if its runtime would not be so high.



It is also interesting to compare the balanced F1 score between test and train sets. If we look at both Random Forest and LGBM, there is a 25% increase from test to train, which means that overfitting occurs during the training process for these classifiers. There are several methods to combat overfitting, so it is possible to tackle this problem; this means that the test and train scores will come towards an average of 0.75-0.8.



Decision

We will continue analyzing the bucket classifier part of the pipeline since it is directly linked to the model's performance. The main plan is to look into more types of classifiers and try to implement the most promising ones. For now, we will focus on improving the classification capabilities of LGBM and Random Forest configurations.