Creator **Olusanmi Hundogan** Created **Dec 18, 2023, 17:21** Last updated **Mar 01, 2024, 15:57**

Files

- notebooks/analyze_stacking_regressor_vs_voting_regressor.py
- local/eval_results/experiment_stacking_regressor_vs_voting_regressor.csv
- experiments/experiment_stacking_regressor_vs_voting_regressor.py

Motivation

The regression method that we currently use for the BucketRegressor is based on voting. A **voting regressor** is an ensemble meta-estimator that fits several base regressors, each on the whole dataset. Then it provides a **weighted** average of the individual predictions to form a final prediction.

While this regression method proved efficient in the past, we wanted to try a new formula: a **stacking regressor**. The primary idea of stacking is to feed the predictions of numerous base models into a higher-level model known as the meta-model or blender, which then combines them to get the final forecast. The architectural advantage of this model is that the stacking process can happen on multiple layers: the blender of the first layer becomes the final model of the second layer etc.

Sklearn provides this implementation of the stacking regressor: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.StackingRegressor.html

Here is an example of generalised (multi-layered) stacking: https://scikit-learn.org/stable/auto_examples/ensemble/plot_stack_predictors.html#sphx-glr-auto-examples-ensemble-plot-stack-predictors-py

Design

Multiple configurations are possible for this stacking type regressor, so we chose two of them which had the greatest potential to provide an accurate comparison with the voting type regressors that were already implemented.

For the implementation of the **simple** stacking regressor we used the following model hierarchy:

Base:

- decision tree
- knn
- ridge + kernel approximation (nystroem with rbf)
- sgd + kernel approximation (nystroem with rbf)

Final:

- gradient boosting

For the implementation of the **double layered** stacking regressor we used the following model hierarchy:

Base

- decision tree
- knn
- ridge + kernel approximation (nystroem with rbf)
- sgd + kernel approximation (nystroem with rbf)

Final:

- stacking regressor using:

  Base

  - extra trees
  - svr + kernel approximation (nystroem with rbf)
  - random forest

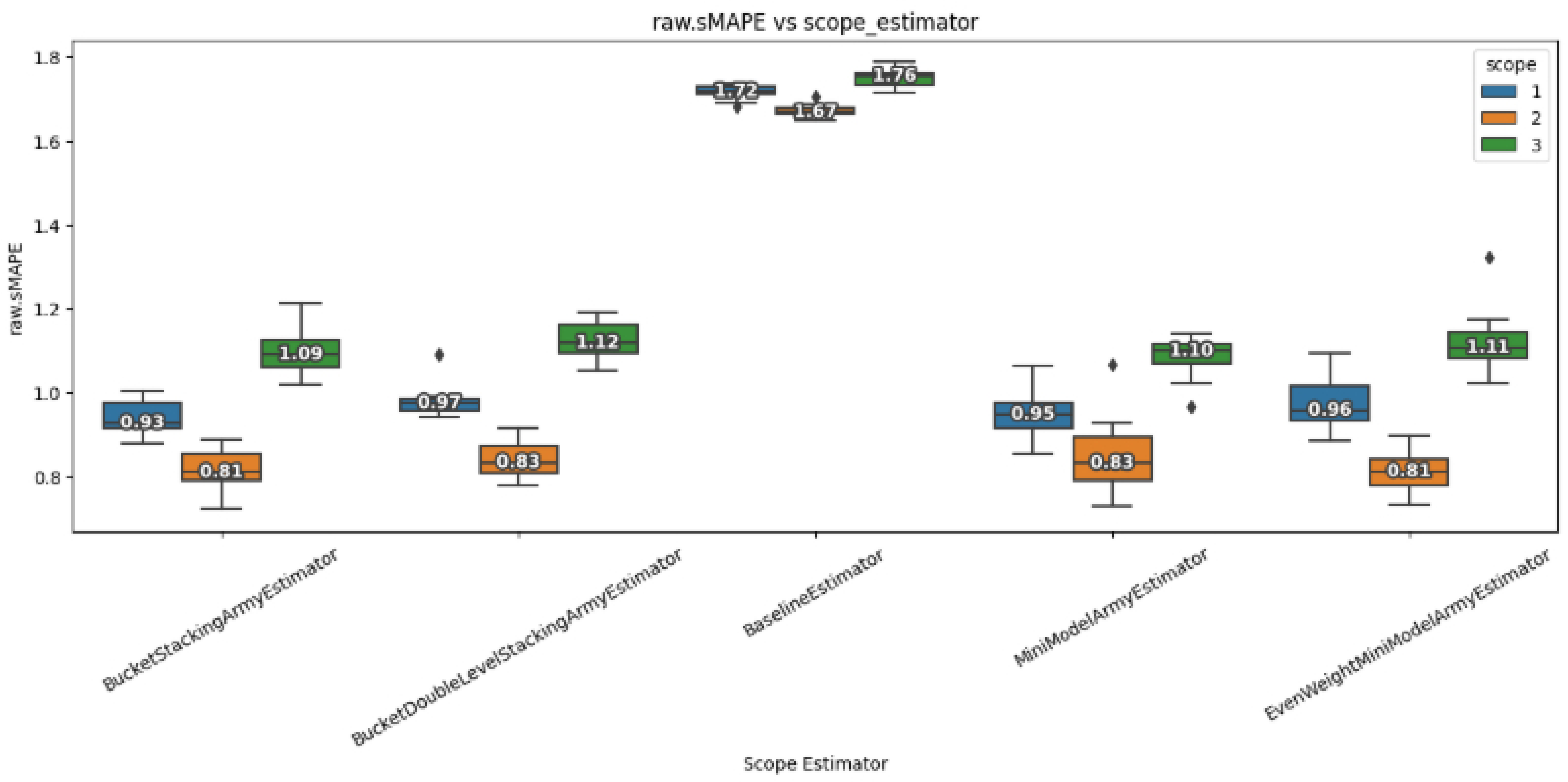
  Final:

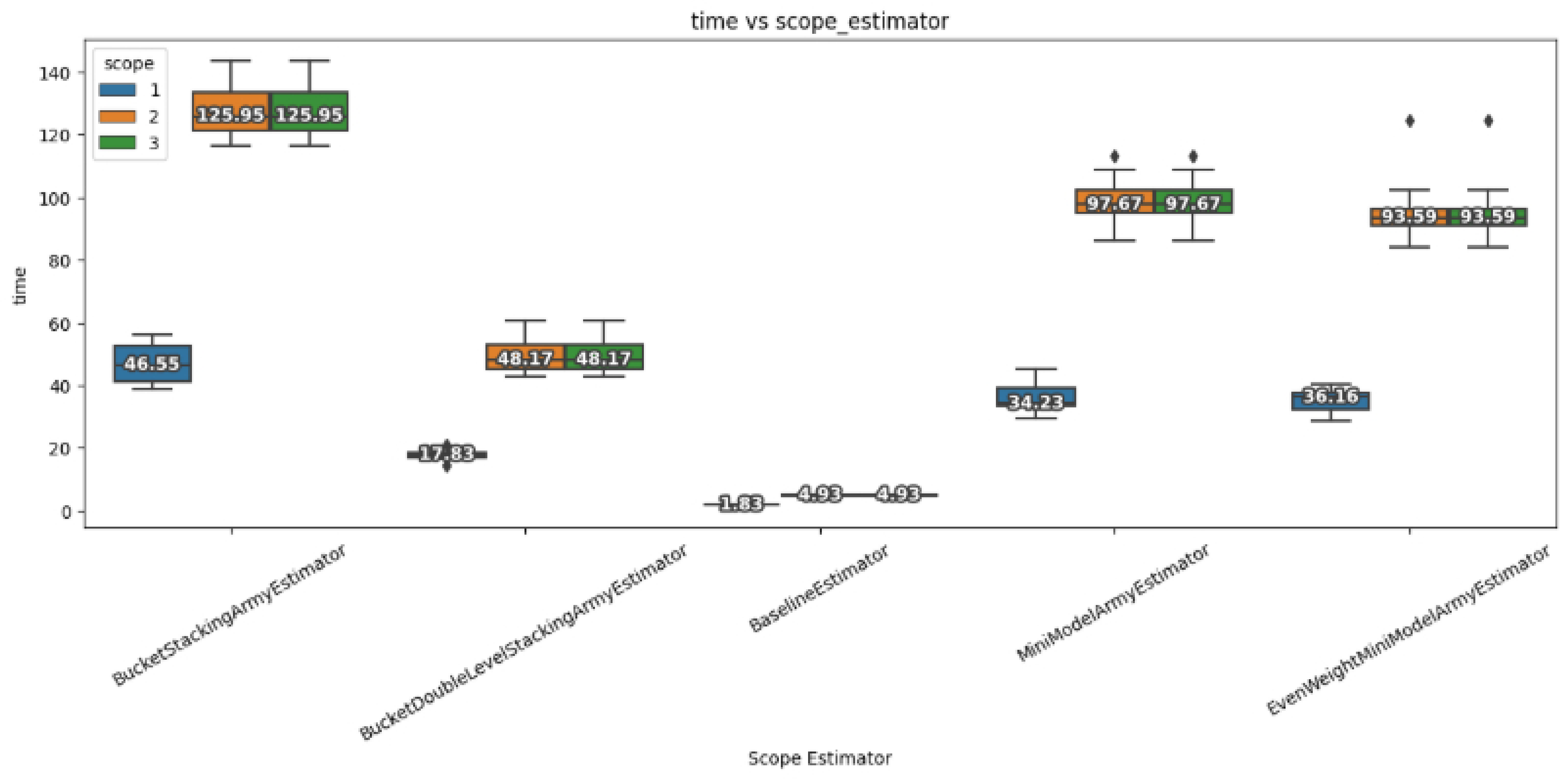
  - lgbm

The experiment compares effects on the overall model estimator when using the voting regressor, the even weighted voting regressor, the simple stacking regressor, the double layered stacking regressor, and the baseline one. There are 10 iterations x 5 models, each having a separate data split.

Results and Insight

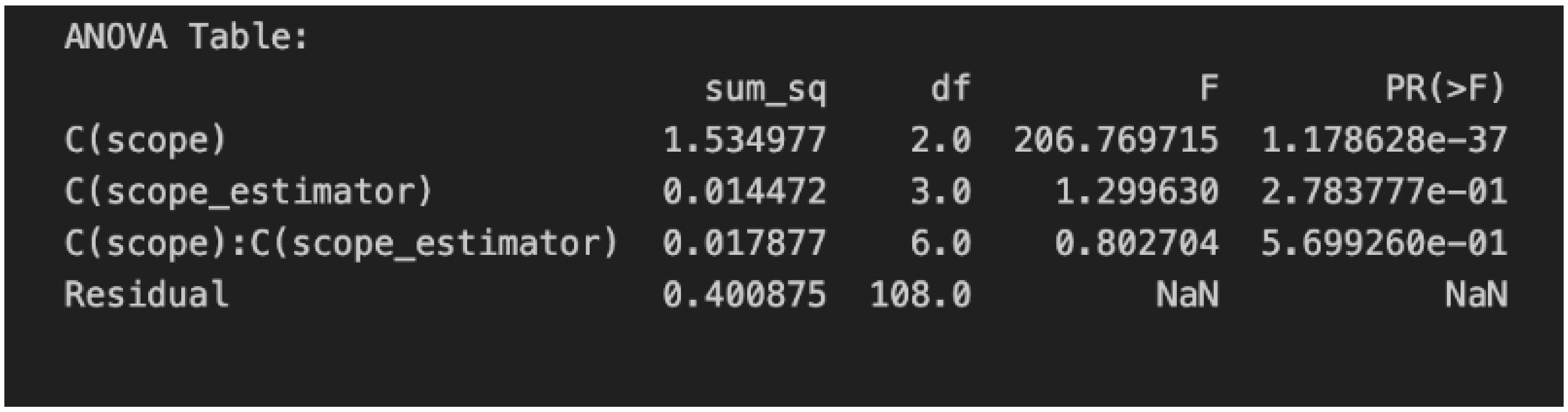
Results show that the estimator which uses an even weight voting mechanism for bucket regression (the EvenWeightMiniModelArmyEstimator) still performs best overall. It turns out that the simple stacking strategy improves the sMAPE performance metric by 3% for scope 1 predictions and 2% for scope 3, but the runtime overhead makes it less worthy in being considered as a replacement for voting.





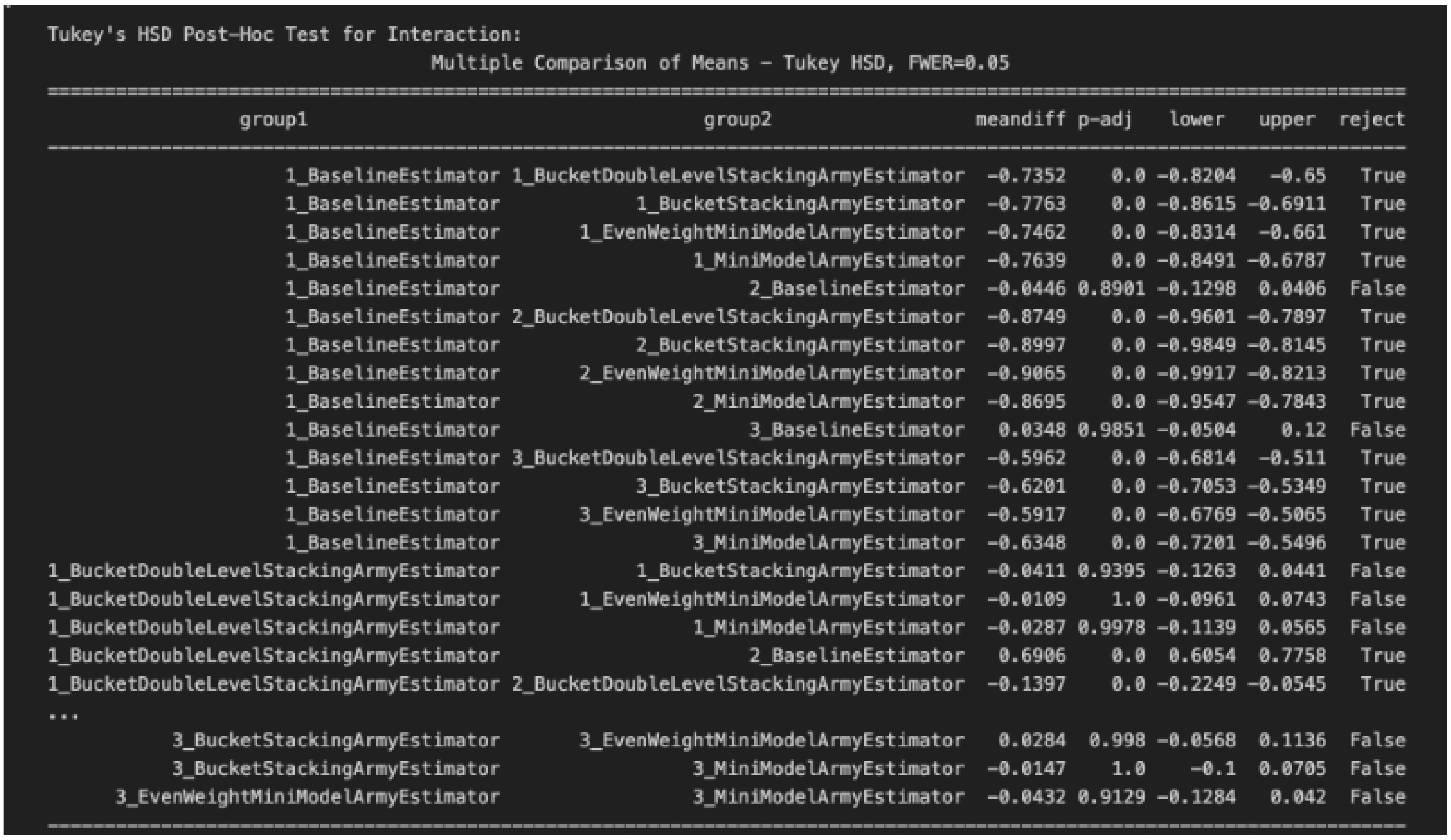
We decided to also perform some statisticsl analysis covering the interaction between scopes and the estimators evaluated above.

The ANOVA table presents a variance analysis between scopes, estimators, and pairs of them. The interaction term between "scope" and "scope_estimator" explores whether the combination of these two variables has a significant impact. The F-statistic is relatively small, and the p-value (5.699260e-01) is larger than 0.05, indicating that the interaction may not be statistically significant.



```
ANOVA Table:
                               sum_sq     df           F        PR(>F)
C(scope)                     1.534977    2.0  206.769715  1.178628e-37
C(scope_estimator)           0.014472    3.0    1.299630  2.783777e-01
C(scope):C(scope_estimator)  0.017877    6.0    0.802704  5.699260e-01
Residual                     0.400875  108.0         NaN           NaN
```

For more details, we decided to look at interactions between specific groups with Tukey's HSD post-hoc test. Unfortunately, the only significant pair results (with p < 0.05) are the ones with the baseline as a member of the pair, or the ones that compare the same estimator but for different scopes, which was expected.



```
Tukey's HSD Post-Hoc Test for Interaction:
                                 Multiple Comparison of Means - Tukey HSD, FWER=0.05
=================================================================================================================
                group1                                  group2                 meandiff p-adj   lower   upper  reject
-----------------------------------------------------------------------------------------------------------------
                    1_BaselineEstimator 1_BucketDoubleLevelStackingArmyEstimator  -0.7352    0.0 -0.8204   -0.65   True
                    1_BaselineEstimator           1_BucketStackingArmyEstimator  -0.7763    0.0 -0.8615 -0.6911   True
                    1_BaselineEstimator       1_EvenWeightMiniModelArmyEstimator  -0.7462    0.0 -0.8314  -0.661   True
                    1_BaselineEstimator               1_MiniModelArmyEstimator  -0.7639    0.0 -0.8491 -0.6787   True
                    1_BaselineEstimator                    2_BaselineEstimator  -0.0446 0.8901 -0.1298  0.0406  False
                    1_BaselineEstimator 2_BucketDoubleLevelStackingArmyEstimator  -0.8749    0.0 -0.9601 -0.7897   True
                    1_BaselineEstimator           2_BucketStackingArmyEstimator  -0.8997    0.0 -0.9849 -0.8145   True
                    1_BaselineEstimator       2_EvenWeightMiniModelArmyEstimator  -0.9065    0.0 -0.9917 -0.8213   True
                    1_BaselineEstimator               2_MiniModelArmyEstimator  -0.8695    0.0 -0.9547 -0.7843   True
                    1_BaselineEstimator                    3_BaselineEstimator   0.0348 0.9851 -0.0504    0.12  False
                    1_BaselineEstimator 3_BucketDoubleLevelStackingArmyEstimator  -0.5962    0.0 -0.6814  -0.511   True
                    1_BaselineEstimator           3_BucketStackingArmyEstimator  -0.6201    0.0 -0.7053 -0.5349   True
                    1_BaselineEstimator       3_EvenWeightMiniModelArmyEstimator  -0.5917    0.0 -0.6769 -0.5065   True
                    1_BaselineEstimator               3_MiniModelArmyEstimator  -0.6348    0.0 -0.7201 -0.5496   True
1_BucketDoubleLevelStackingArmyEstimator           1_BucketStackingArmyEstimator  -0.0411 0.9395 -0.1263  0.0441  False
1_BucketDoubleLevelStackingArmyEstimator       1_EvenWeightMiniModelArmyEstimator  -0.0109    1.0 -0.0961  0.0743  False
1_BucketDoubleLevelStackingArmyEstimator               1_MiniModelArmyEstimator  -0.0287 0.9978 -0.1139  0.0565  False
1_BucketDoubleLevelStackingArmyEstimator                    2_BaselineEstimator   0.6906    0.0  0.6054  0.7758   True
1_BucketDoubleLevelStackingArmyEstimator 2_BucketDoubleLevelStackingArmyEstimator  -0.1397    0.0 -0.2249 -0.0545   True
...
           3_BucketStackingArmyEstimator       3_EvenWeightMiniModelArmyEstimator   0.0284  0.998 -0.0568  0.1136  False
           3_BucketStackingArmyEstimator               3_MiniModelArmyEstimator  -0.0147    1.0    -0.1  0.0705  False
      3_EvenWeightMiniModelArmyEstimator               3_MiniModelArmyEstimator  -0.0432 0.9129 -0.1284   0.042  False
-----------------------------------------------------------------------------------------------------------------
```

We also tried the Mixed Linear Model (MixedLM) regression analysis. This type of model is used when dealing with clustered or hierarchical data where observations are not independent. In this case, there are 12 groups, and each group has 10 observations. The dependent variable is "raw_sMAPE," and the independent variable is "scope_estimator" with multiple levels. The model considers both fixed effects (overall relationships) and random effects (variance within groups).

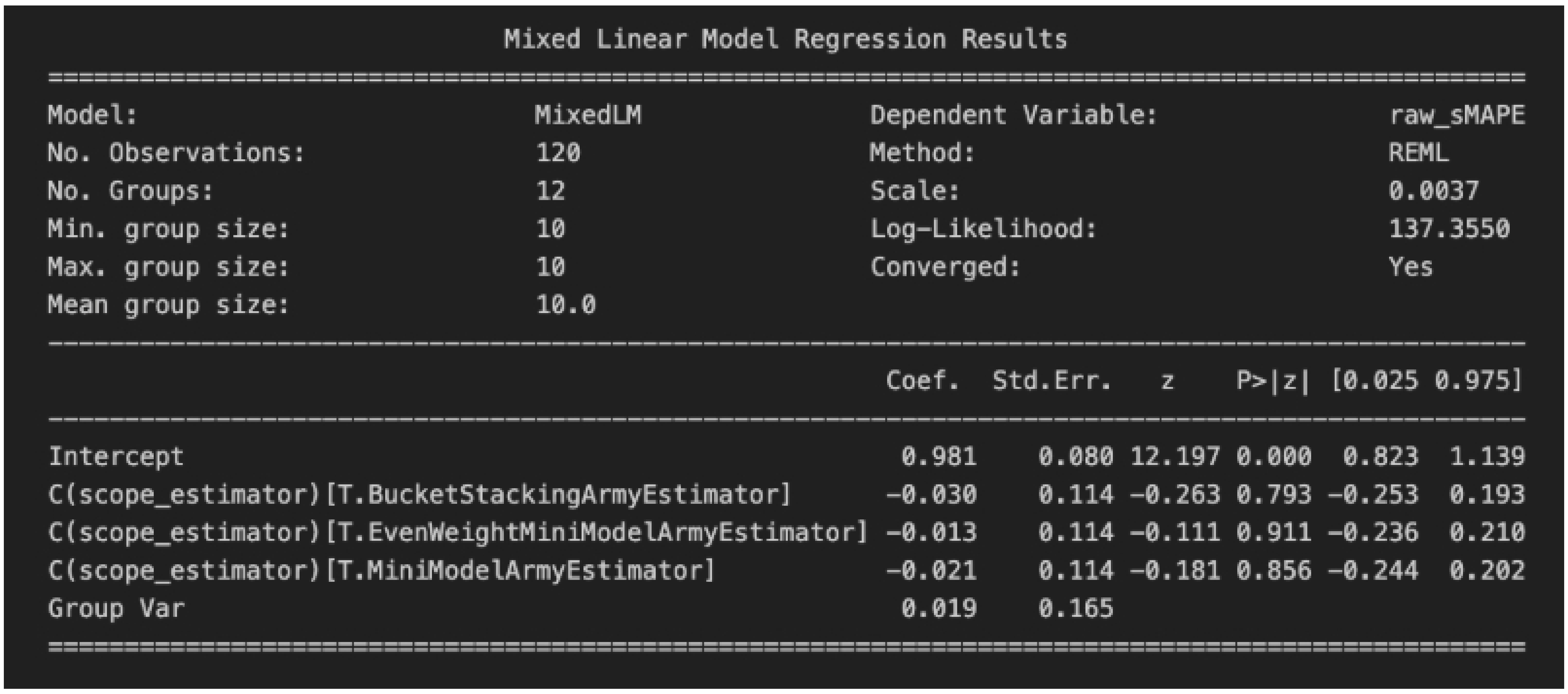
Fixed effects:

The "coef" column for C(scope_estimator) shows the coefficients for the different levels of the 'scope_estimator' variable. Each coefficient represents the estimated change in the dependent variable when the corresponding level of 'scope_estimator' is compared to the reference level ('BaselineEstimator' in this case). All p-values are below 0.05, suggesting that the variable is not statistically significant. For example:

- 'BucketStackingArmyEstimator': -0.030 (p-value = 0.793)
- 'EvenWeightMiniModelArmyEstimator': -0.013 (p-value = 0.911)
- 'MiniModelArmyEstimator': -0.021 (p-value = 0.856)

Random effects:

Group Var represents the estimated variance of the random effects associated with different groups formed by the interaction of 'scope' and 'scope_estimator.' In this case, the estimated variance is 0.019 with a standard error of 0.165.



```
                        Mixed Linear Model Regression Results
=================================================================================================
Model:                          MixedLM               Dependent Variable:               raw_sMAPE
No. Observations:               120                   Method:                           REML
No. Groups:                     12                    Scale:                            0.0037
Min. group size:                10                    Log-Likelihood:                   137.3550
Max. group size:                10                    Converged:                        Yes
Mean group size:                10.0
-------------------------------------------------------------------------------------------------
                                                       Coef.  Std.Err.   z    P>|z| [0.025 0.975]
-------------------------------------------------------------------------------------------------
Intercept                                               0.981    0.080 12.197 0.000  0.823  1.139
C(scope_estimator)[T.BucketStackingArmyEstimator]      -0.030    0.114 -0.263 0.793 -0.253  0.193
C(scope_estimator)[T.EvenWeightMiniModelArmyEstimator] -0.013    0.114 -0.111 0.911 -0.236  0.210
C(scope_estimator)[T.MiniModelArmyEstimator]           -0.021    0.114 -0.181 0.856 -0.244  0.202
Group Var                                               0.019    0.165
=================================================================================================
```

Decision

We will continue using the even weighted voting mechanism because it produces the best performance for the least runtime so far. Further cascading investigations on how much interaction we have between scopes and different estimator configurations might be a good idea in order to improve the overall model performance.