

Combine Signals for Enhanced Alpha

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Sensational Learner,
I appreciate your effort and hard work that you had put in. I'm glad to see you succeed. Congratulations and wish you good luck for your continued success.

Features and Labels

✓

Describe the relationship between the shifted labels.

Correct!

Indeed, in any case, the next day's returns are highly correlated compared to the present day.

✓

Correctly implement the `train_valid_test_split` function.

Awesome job!

The `train_valid_test_split` is correctly implemented to split the dataset into the train, validation, and test datasets. This is good work! ✓

Pro Tip

You check out the following for more.

• [How to split your dataset to train and test datasets using SciKit Learn](#)

Rate this review

★

★

★

★

★

Random Forests

✓

Describe why dispersion_20d has the highest feature importance, when the first split is on the Momentum_1YR feature.

Good observation of how feature importance is measured!
Moreover, the `dispersion_20d` feature has more information gain when dealing with more splits.

Check [information gain](#) and [entropy](#)

✓

Describe how the accuracy changes over time and what indicates the model is overfitting or underfitting.

Good job to observe from the `Random Forest Accuracy` plot that there is an insignificant change in accuracy after a few hundred trees.
Right justification as the model is overfitting with the validation accuracy dropping.

Pro Tips

• [Overfitting and Underfitting With Machine Learning Algorithms](#)

• [Overfitting vs. Underfitting: A Complete Example](#)

• [How do we detect overfitting and underfitting in Machine Learning?](#)

Overlapping Samples

✓

Correctly implement the `non_overlapping_samples` function.

Nice work implementing the `non_overlapping_samples` function, by dropping all overlapping samples from the dataset. ✓

✓

Correctly implement the `bagging_classifier` function.

Nicely done!

Good job correctly building the bagging classifier. 🎉 +1:

✓

Correctly implement the `calculate_oob_score` function.

Great work here getting the correct OOB scores.

✓

Correctly implement the `non_overlapping_estimators` function.

Good job looping through the classifiers and fitting the model. 🙌

DOWNLOAD PROJECT

RETURN TO PATH